

UNIVERSITATEA POLITEHNICA BUCUREȘTI  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE



## PROIECT DE DIPLOMĂ

PoliHub

Răchițeanu Robert-Alexandru

**Coordonator științific:**  
Prof. dr. ing. Alexandru Boicea

**BUCUREȘTI**

2023

UNIVERSITY POLITEHNICA OF BUCHAREST  
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS  
COMPUTER SCIENCE DEPARTMENT



## DIPLOMA PROJECT

PoliHub

Răchițeanu Robert-Alexandru

**Thesis advisor:**

Prof. dr. ing. Alexandru Boicea

**BUCHAREST**

2023

# CUPRINS

Sinopsis .....	4
Abstract.....	4
1    Introducere.....	5
1.1    Context.....	5
1.2    Problema .....	5
1.3    Obiective .....	6
1.4    Structura lucrării .....	6
2    Analiza și specificarea cerințelor .....	7
3    Studiu de piață / Abordări existente.....	8
3.1    Automatic Attendance System Using Biometric Authentication and Varying QR Code.....	8
3.2    IoT based smart attendance monitoring system using RFID.....	9
4    Soluția propusă.....	11
4.1    Arhitectura Sistemului.....	11
4.2    Tehnologii Utilizate .....	14
4.2.1    React .....	14
4.2.2    TypeScript .....	15
4.2.3    .NET .....	15
4.2.4    PostgreSQL.....	16
4.2.5    Docker.....	17
4.3    Diagrama Bazei de Date .....	17
4.4    Tipuri de Utilizatori .....	22
5    Detalii de implementare.....	23
5.1    Inițializarea Proiectului .....	23
5.2    Funcționalități .....	24
5.2.1    Operații CRUD.....	24
5.2.2    Bara de Navigare.....	25
5.2.3    Autentificare .....	25
5.2.4    Pagina Principală.....	27
5.2.5    Vizualizarea Orarului.....	28

5.2.6	Vizualizarea Prezențelor .....	28
5.2.7	Vizualizarea Materiilor, Cursurilor și a Laboratoarelor .....	29
5.2.8	Înregistrarea Prezențelor .....	31
5.2.9	Gestionarea Datelor de către Administrator .....	32
5.3	Aspecte de Securitate .....	33
6	Studiu de caz / Evaluarea rezultatelor.....	35
7	Concluzii.....	36
8	Bibliografie.....	37
9	Anexe .....	39

## CUPRINS FIGURI

Figura 1: Pagina principală pentru profesori (Checkit).....	8
Figura 2: Interfața utilizatorului pentru a vedea prezențele înregistrate (Checkit).....	10
Figura 3: Three-tier architecture.....	11
Figura 4: Multi Page Application design.....	13
Figura 5: VDOM.....	14
Figura 6: TechEmpower benchmark.....	15
Figura 7: Bază de date relațională.....	16
Figura 8: Diagrama bazei de date.....	18
Figura 9: Tabela User.....	18
Figura 10: Tabela Subject.....	19
Figura 11: Tabela Course.....	19
Figura 12: Tabela Course Instance.....	20
Figura 13: Tabela Laboratory.....	20
Figura 14: Tabela Laboratory Instance.....	21
Figura 15: Tabela Subject User.....	21
Figura 16: Fișierul appsettings.json.....	23
Figura 17: Bara de navigare.....	25
Figura 18: Pagina de înregistrare.....	26
Figura 19: Pagina de autentificare.....	26
Figura 20: Pagina principală.....	27
Figura 21: Pagina Orar.....	28
Figura 22: Pagina Prezențe.....	29
Figura 23: Pagina Materie.....	30
Figura 24: Formular pentru adăugarea unui curs.....	31
Figura 25: Pagina Instanță de Laborator.....	32
Figura 26: Pagina Materii.....	33
Figura 27: JWT pentru administrator.....	34

## **SINOPSIS**

PoliHub este un sistem de management al prezențelor la ore, ce vine în ajutorul atât al studenților cât și al profesorilor prin eficientizarea procesului didactic. Aplicația folosește un cod QR distinct pentru fiecare curs și laborator, pe care studenții îl folosesc pentru a-și înregistra prezența și a obține acces la un portal web specializat. Totodată, platforma le oferă utilizatorilor posibilitatea de a-și vedea orarul într-un mod intuitiv, fapt ce duce la o gestiune mai bună a timpului.

## **ABSTRACT**

PoliHub is a class attendance management system that aids both students and teachers by streamlining the educational process. The application uses a distinct QR code for each course and laboratory, which students use to register their attendance and gain access to a specialized web portal. Furthermore, the platform offers users the ability to view their schedule in an intuitive way, leading to better time management.

# **1 INTRODUCERE**

## **1.1 Context**

Educația și învățarea reprezintă unele dintre cele mai importante aspecte în dezvoltarea umană și în cea a societății. Sectorul educațional se concentrează în prezent pe găsirea de noi metode de îmbunătățire a experienței de învățare, simplificarea sarcinilor administrative și utilizarea cât mai eficientă a resurselor disponibile, toate în contextul transformării digitale.

Managementul prezenței este o componentă crucială a sistemului educațional. Acesta a fost tradițional un proces laborios și predispus la erori, bazându-se frecvent pe tehnici manuale sau digitale de bază. Urmărirea eficientă a prezenței este esențială pentru îmbunătățirea eficacității generale a instituțiilor de învățământ. Pentru a realiza acest lucru, este esențial să se creeze o soluție intuitivă, sigură și sofisticată.

Ideea principală a proiectului este de a crea coduri QR individuale pentru fiecare curs și laborator. Aceste coduri sunt apoi utilizate de studenți pentru a înregistra prezența prin scanarea lor cu telefonul mobil. Implementarea acestui proces are multiple beneficii, inclusiv simplificarea urmăririi prezenței, reducerea probabilității de erori cauzate de introducerea manuală, îmbunătățirea securității datelor și diminuarea sarcinii de lucru pentru instructori.

## **1.2 Problema**

Instituțiile de învățământ au fost limitate mult timp de dificultățile și ineficiențele metodelor tradiționale de monitorizare a prezenței. Obiectivul subcapitolului privind este de a explora în profunzime aceste probleme, enunțând punctele slabe pe care PoliHub intenționează să le rezolve prin implementarea unui sistem inteligent de gestionare a prezenței bazat pe coduri QR.

Metodele tradiționale de administrare a prezenței implică foile de prezență fizice pe care studenții se trec manual cu pixul sau foile de calcul gestionate direct de profesori, care sunt consumatoare de timp, ineficiente și posibil cauzatoare de erori. Acest timp ar fi mai bine utilizat pentru îmbunătățirea experienței de învățare a studenților sau pentru implicarea în alte demersuri pedagogice. Proiectul facilitează procesul de monitorizare a prezenței, reducând sarcina administrativă asupra cadrelor didactice și îmbunătățind eficiența generală a clasei.

Monitorizarea manuală a prezenței este susceptibilă la erori umane, inclusiv intrări incorecte, omisiuni și chiar manipulări intenționate din partea studenților. Aceste lacune ale sistemului original pot duce la date de prezență nesigure, care pot avea implicații de lungă durată pentru evaluarea performanței studenților și asigurarea integrității academice. Prin automatizarea procesului de prezență cu coduri QR unice, aplicația reduce posibilitatea erorilor umane și crește precizia datelor.

Gestionarea prezenței pe bază de hârtie contribuie la risipa excesivă de hârtie și aduce consecințe nefavorabile mediului înconjurător. Înlocuind metodele tradiționale cu o soluție digitală, „PoliHub” nu doar facilitează procesul de administrare a prezenței, dar și reduce impactul asupra mediului asociat cu consumul de hârtie.

### **1.3 Obiective**

Obiectivul principal al proiectului este de a crea un sistem inteligent, eficient și intuitiv de administrare a prezenței, care să depășească dezavantajele și limitările metodelor convenționale actuale.

Intenția este de a proiecta, implementa și evalua un sistem de monitorizare a prezenței scalabil și robust, capabil să răspundă nevoilor instituțiilor de dimensiuni diferite.

De asemenea, aplicația urmărește să maximizeze potențialul sistemului pentru o adoptare extinsă, garantând adaptabilitatea proiectului la diverse medii educaționale și totodată intenționând să eficientizeze procesul de gestiune a prezenței în instituțiile de învățământ și să promoveze un mediu de învățare mai eficace.

### **1.4 Structura lucrării**

În capitolul 2 se va prezenta analiza cerințelor, detaliind modul în care acestea au fost selectate, fiind de asemenea specificate care sunt cerințele alese, în urma consultării unor persoane din domeniu.

Capitolul 3 prezintă studiul de piață realizat pentru analiza abordărilor existente, realizându-se totodată o comparație între acestea și soluția propusă în lucrare.

„Soluția propusă” este capitolul 4 al lucrării și creează o perspectivă generală a modului de abordare a aplicației, explicând arhitectura sistemului, tehnologiile utilizate, structura bazei de date și tipul de utilizatori, dar totodată avantajele aduse de acestea și motivul alegerii lor.

În capitolul 5 este descris în amănunt modalitatea prin care au fost întrunite cerințele utilizatorilor, demonstrând totodată capacitățile sistemului propus prin prezentarea funcționalităților aplicației.

Capitolul 6 arată modalitatea de evaluare a sistemului, precizând atât opiniile grupului țintă în legătură cu întrunirea cerințelor funcționale și nefuncționale de către proiect, cât și posibilele adăugări care s-ar putea aduce acestuia.

Ultimul capitol, anume capitolul 7, are rolul de a trage concluziile finale asupra implementării proiectului, capitolele 8 și 9 fiind bibliografia, respectiv anexele lucrării.



## 2 ANALIZA ȘI SPECIFICAREA CERINȚELOR

Un rol fundamental în crearea unui anumit produs este jucat, în partea de cercetare și dezvoltare, de explicitarea cerințelor clientului, analiza și implementarea produsului în funcție de acestea. Cu ajutorul lor, dezvoltatorii determină atât structura sistemului (arhitectura, tehnologiile utilizate, organizarea bazei de date), cât și funcționalitățile care trebuie implementate pentru a satisface așteptările utilizatorilor.

În urma unei analize a pieței și a interacțiunii atât cu mai mulți studenți din cadrul universitar, cât și cu expertiza unui profesor, au fost create o serie de cerințe atât funcționale, cât și nefuncționale, punând pe primul loc nevoile beneficiarilor. Astfel, în funcție de rolul pe care aceștia îl joacă în procesul aplicației, cerințele sunt împărțite în:

- Cerințele studenților:
  - Înregistrarea prezențelor – studenții doresc să aibă o modalitate facilă de înregistrare a prezenței, dar și eficientă din punct de vedere al timpului consumat;
  - Vizualizarea prezențelor – studenții vor o platforma prin care își pot vedea într-un mod intuitiv prezențele la materiile lor;
  - Vizualizarea orarului – studenții doresc să aibă un mod de a-și vedea orarul pentru a-și putea gestiona mai bine timpul lor;
  - Schimbări de orar – studenții vor să fie la curent cu ultimele schimbări de orar și să aibă un mod în care să vadă dacă un laborator sau un curs s-a anulat.
- Cerințele profesorilor:
  - Înregistrarea prezențelor – profesorii doresc să aibă o modalitate prin care să poată să înregistreze studenții care participă la cursurile și laboratoarele pe care le susțin;
  - Gestiunea prezențelor – profesorii își doresc un mod prin care să poată vizualiza prezențele studenților la laboratoare sau cursuri și care să fie ușor de folosit;
  - Vizualizarea orarului – profesorii, la fel ca și studenții vor să aibă un mod de a-și vizualiza orarul pentru a avea o perspectivă mai bună asupra programului lor.
- Cerințe pentru administrator:
  - Operații CRUD pe toate datele – administratorul are nevoie de permisiuni pentru a realiza acțiuni de creare (create), citire (read), actualizare (update), ștergere (delete) pe toate datele din sistem;
  - Acces facil asupra datelor – administratorul are nevoie de o modalitate de a accesa ușor informațiile din baza de date pentru manipularea lor.
- Cerințe pentru toți utilizatorii:
  - Securitatea datelor – toți utilizatorii își doresc ca datele lor să fie protejate de un sistem care le asigură securitatea datelor;
  - Interfață intuitivă – toți utilizatorii vor o platformă cu un design intuitiv, ușor de înțeles pentru a ajunge mai rapid la datele relevante pentru ei.

### 3 STUDIU DE PIAȚĂ / ABORDĂRI EXISTENTE

Având în vedere cerințele enunțate în secțiunea precedentă, în acest capitol se vor analiza comparativ câteva dintre soluțiile similare din întreaga lume, deja existente în domeniu, cu soluția propusă în această lucrare.

Astfel, după o cercetare a lucrărilor din această arie, s-au găsit următoarele proiecte:

3.1 Automatic Attendance System Using Biometric Authentication and Varying QR Code [1]

3.2 IoT based smart attendance monitoring system using RFID [2]

#### 3.1 Automatic Attendance System Using Biometric Authentication and Varying QR Code [1]

În articolul elaborat de Aishwarya Varadannanavar, Anisha Nair și Amit Kuma se prezintă soluția unui sistem integrat numit „Checkit” care combină autentificarea cu date biometrice, anume recunoaștere facială, cu cea de cod QR generat dinamic, pentru asigurarea securității în momentul realizării prezenței.

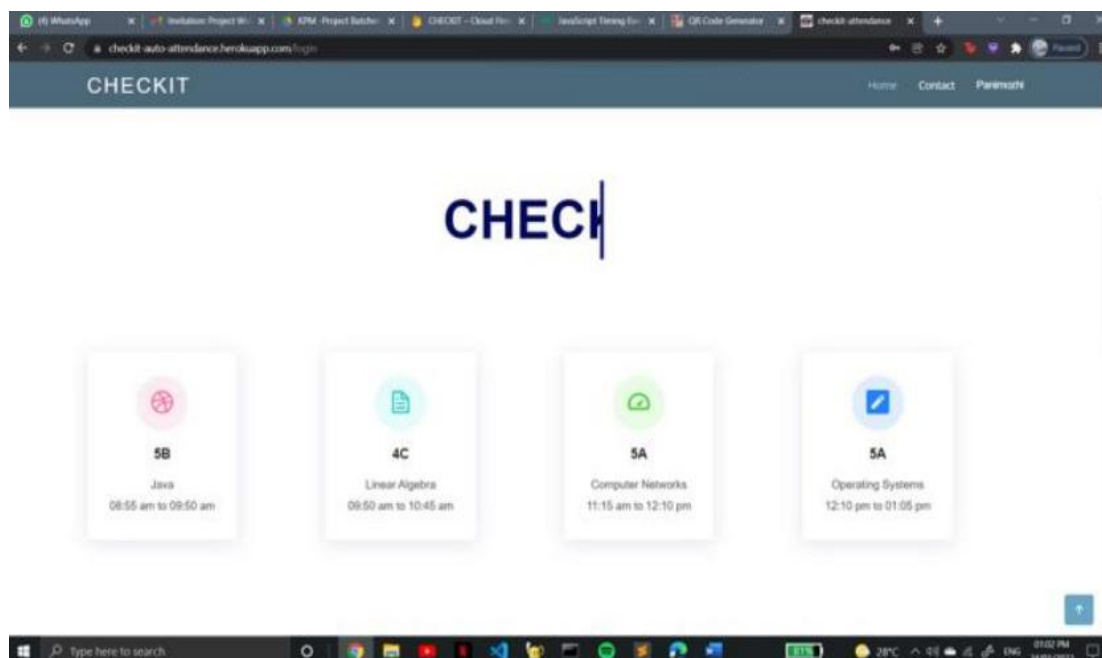


Figura 1: Pagina principală pentru profesori (Checkit) [1]

Problematica ce a stat la baza acestui proiect a fost reducerea limitărilor sistemelor convenționale, cum ar fi erorile manuale sau reducerea timpului necesar pentru prezență, întrucât după analizele lor, se pierde minim 5 minute în cadrul fiecărei ore doar pentru efectuarea prezenței.

Datorită mecanismului inteligent de autentificare pe două niveluri, autorii articolului susțin că sistemul dezvoltat de ei eficientizează combaterea acțiunilor frauduloase. Astfel, se asigură prezența fizică a studenților atât prin modelul de recunoaștere facială bazat pe algoritmi de precizie foarte bună, cât și prin codul QR pe care aceștia îl scanează în sala de curs.

În partea de front-end a sistemului, se află interfața grafică a aplicației ce scanează fețele studenților și permite profesorilor să creeze instanțe de ore pentru materiile lor. Odată ajunse în back-end, datele biometrice ale studenților, cât și codurile QR generate sunt reținute cu ajutorul unor framework-uri de deep learning, realizate prin rețele neuronale convoluționale prin care se verifică autenticitatea datelor.

Una dintre cele mai mari inovații ale acestui proiect este cuplarea datelor biometrice cu codul QR care, generat de mai multe ori, acesta reține data scanării lui și astfel studenții nu pot scana codul, decât în intervalul orar aferent orei respective.

Această lucrare prezintă o soluție sofisticată și avansată tehnologic în legătură cu o problemă comună cu care se confruntă sistemul educațional. Prin asocierea codurilor QR generate dinamic cu recunoașterea facială, sistemul produs de autorii acestui articol au reușit să aducă o rezolvare eficientă a problemei de realizare a prezenței în sălile de curs, minimizând acțiunile dolosive ale studenților în acest proces.

### **3.2 IoT based smart attendance monitoring system using RFID [2]**

Lucrarea redactată de U. Koppikar et al. explorează sisteme de tip Internet of Things (IoT) și Radio Frequency Identification (RFID) în contextul modelelor de management al prezenței atât în domeniul educației, cât și în cel profesional pentru a îmbunătăți eficacitatea și precizia.

Cercetând tehnologii din mai multe contexte precum cele ale sălilor de clasă, localuri de evenimente sau cel al locului de muncă, autorii au descoperit că utilizarea cardurilor RFID aduce un beneficiu deosebit, acela de eficiență mai ridicată a realizării prezențelor la un cost mai scăzut, așa cum enunță și Sumita Nainan et al. în „RFID Technology Based Attendance Management System”[3].

Din punct de vedere tehnic, sistemul propus de lucrare vizează componente hardware precum un cititor de carduri RFID, Arduino UNO și LED-uri, conectate la un computer pentru a asigura conexiunea datelor și prelucrarea acestora. Pentru a se înregistra ca fiind prezent, utilizatorul trebuie să își folosească propriul card RFID, pe care îl scanează la cititorul de carduri, iar apoi în funcție de validitatea cardului, dispozitivul aprinde LED-ul verde pentru confirmare sau roșu pentru rejectare.

Utilizatorul trebuie să fie înregistrat în prealabil în cadrul sistemului, acesta primind odată cu crearea cardului RFID un număr ce definește ID-ul său de student, angajat, etc.

După ce se scanează cardul, sistemul verifică timpii la care s-a realizat această acțiune și în cazul în care aceștia corespund cu timpii interni de venire și de plecare din locație, utilizatorul este trecut ca fiind prezent sau nu.

Mai mult decât atât, sistemul vine integrat cu o interfață grafică implementată printr-o aplicație web ce rulează pe un server, care face procesul mai intuitiv pentru administrator, întrucât poate avea acces la date de oriunde.

RFID NUMBER	01-03-2019	02-03-2019	03-03-2019	04-03-2019	05-03-2019	06-03-2019	07-03-2019	08-03-2019	09-03-2019	10-03-2019
44 99 53 26	08:25:25	09:25:25	10:25:25	10:25:25	08:30:25	09:25:25	09:05:25	10:25:25	09:25:25	09:45:25
44 99 53 25	09:25:25	09:00:25	10:00:25	10:00:25	09:30:25	09:10:25	09:11:25	10:23:25	09:35:25	09:55:25

Figura 2: Interfața utilizatorului pentru a vedea prezențele înregistrate (Checkit) [2]

Nu în ultimul rând, folosirea tehnologiilor de tip IoT sau RFID, aduc o avansare semnificativă în domeniul digitalizării sistemelor de înregistrare a prezenței, întrucât cresc eficiența de transport a informațiilor, scad posibilitatea erorilor umane și oferă versatilitate, putând fii utilizate atât în sălile de clasă, cât și la locul de muncă sau în sălile de evenimente.

## 4 SOLUȚIA PROPUȘĂ

În cadrul capitolelor precedente au fost descrise motivele necesității aplicației, evidențiând atât contextul cât și problema în cauză. Realizând analize atât asupra cerințelor utilizatorilor consultați, cât și asupra soluțiilor deja existente pe piață, s-a pus la cale un plan de construire a unei aplicații scalabile, care să poată întruni obiectivele propuse.

Astfel, în această parte a lucrării va fi explicată în detaliu structura sistemului, descriind procesul prin care au fost selectate arhitectura sistemului, tehnologiile utilizate, structura bazei de date și tipurile de utilizatori.

### 4.1 Arhitectura Sistemului

Datorită separației dintre nivelurile front-end, back-end și a bazei de date, dar și a puterii de scalabilitate, pentru dezvoltarea sistemului a fost implementată o arhitectură de tip three-tier.

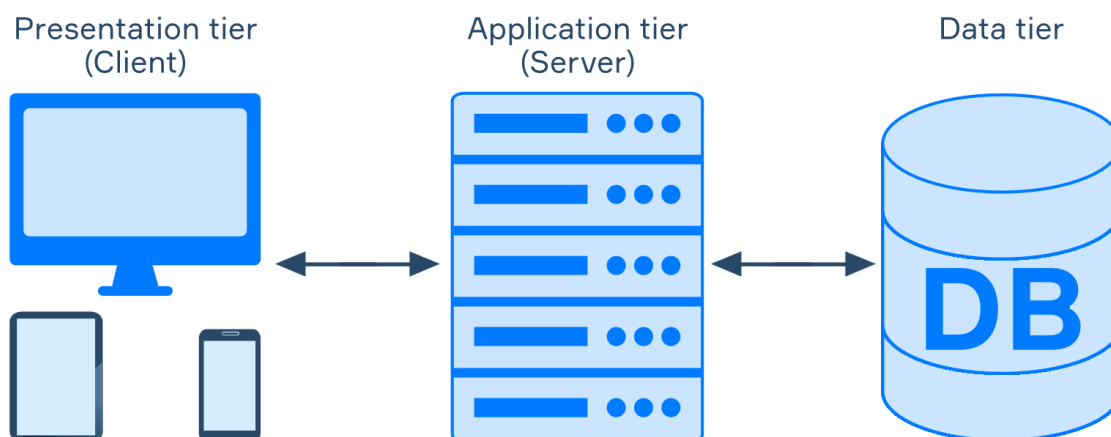


Figura 3: Three-tier architecture <sup>1</sup>

După cum spune și numele, arhitectura este împărțită pe 3 niveluri și anume nivelul de prezentare (front-end), nivelul de logică sau aplicație (back-end) și nivelul de date sau baza de date, fiecare dintre acestea având un rol bine definit în procesul aplicației [4]:

- Nivelul de prezentare, fiind totodată cel mai înalt nivel al sistemului, a fost implementat folosind React și TypeScript. Acest strat este modul prin care utilizatorul obișnuit interacționează cu funcțiile programului. Nivelul vine sub forma unui Graphical User Interface (GUI), accesat din motorul de navigare și care afișează și capturează informațiile pe care utilizatorul le vizualizează/introduce.
- Nivelul de logică este partea de mijloc, ce are rolul de a prelucra datele primite de la nivelul anterior. Practic, această latură a sistemului, integrată folosind .NET, trebuie

---

<sup>1</sup> © <https://hyperskill.org/learn/step/25083>

să intercepteze și să furnizeze informațiile nivelului de prezentare, modelându-le folosind anumite procese după logica aplicației. Totodată, acesta expune o serie de endpoint-uri pe care nivelul de prezentare le accesează cu scopul realizării operațiilor CRUD.

- Nivelul de date are ca scop găzduirea server-ului bazei de date (de tip relațional în acest caz), care utilizează PostgreSQL pe parte de implementare. Acesta este asignat cu gestiunea datelor și are ca și funcționalități înmagazinarea și furnizarea acestora pentru nivelul precedent.

Fiind una dintre cele mai utilizate tipuri de arhitecturi, aceasta aduce o varietate de beneficii [5], dintre care menționăm:

- Scalabilitatea – datorită delimitării exacte a nivelurilor, fiecare dintre acestea pot fi schimbate sau scalate (atât orizontal, cât și vertical) independent de celelalte. Totodată acest tip de arhitectură permite modificarea stivei de tehnologie a unui singur nivel (atât timp cât nu sunt modificate interfețele de comunicare) în funcție de nevoile clientului;
- Fiabilitatea – în cazul unei pene de curent sau unui eveniment ce produce daune asupra unei componente ale sistemului, celelalte niveluri nu sunt atât de probabili să fie afectate;
- Securitatea – întrucât nivelurile de prezentare și cel de date nu interacționează direct, ci prin intermediul celui de logică, sistemul este mai puțin vulnerabil la atacuri ce vizează direct datele (de exemplu SQL injection), back-end-ul funcționând ca un strat de filtrare.

Acum o să ne concentrăm atenția asupra structurii părții de front-end, care este de tip Multi Page Application (MPA), fapt datorat structurii sale intuitive, repartizată în mai multe pagini HTML distincte.

## MPA Design

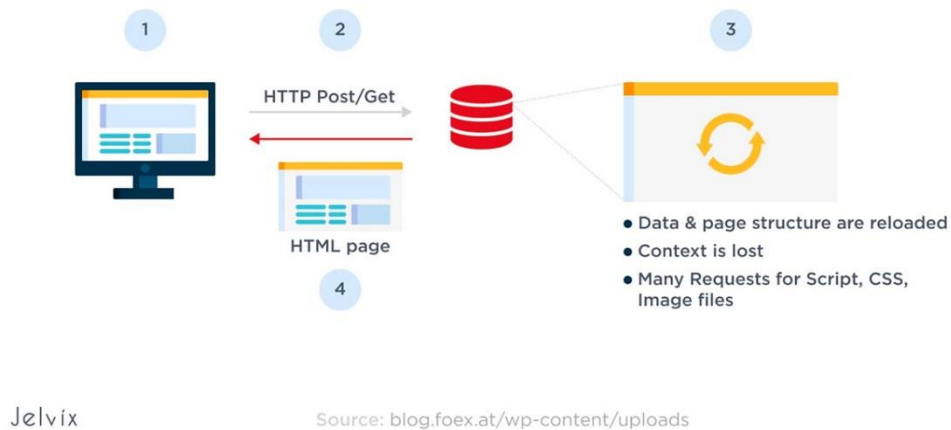


Figura 4: Multi Page Application design <sup>2</sup>

După cum se poate observa și în figura anterioară, acest tip de structură presupune reafișarea paginii la fiecare modificare produsă în front-end sau afișarea uneia noi.

Mai mult decât atât, comunicarea componentelor back-end și front-end, fiind de tip client-server cu interfațare RESTful API, se face cu ajutorul mai multor tipuri de cereri request HTTP trimise de către nivelul prezentare, care primește în schimb cereri response ce atestă statusul prin coduri și întoarce un mesaj. Hypertext Transfer Protocol sau HTTP este un protocol de tip text la nivel de aplicație (în modelul OSI) care exprimă modul de formatare a datelor ce sunt transmise și are în componența sa: tipul metodei, sursa, destinația, antet și opțional un payload și un body (în funcție de tipul metodei). În cadrul aplicației au fost utilizate cereri folosind 4 metode [6]:

- GET – această metodă are rol de citire a datelor, iar în momentul în care este apelată, în cazul în care resursa cerută se află în baza de date, se returnează o reprezentare în format JSON a datelor cerute, alături de un cod de confirmare 200 (OK). În caz alternativ când informația cautată nu a fost identificată, se întoarce codul 404 (NOT FOUND);
- POST – adăgarea unei noi resurse în sistem se face cu ajutorul metodei POST, care primește datele de creare a noii componente, după care trimite codul 201 (CREATED) pentru confirmare;
- PUT – prin acest tip de cerere se realizează actualizarea unei resurse specificate, fiind încapsulate în interiorul cererii câmpurile cu date modificate. La fel ca și în cazul metodei POST, în momentul actualizării cu succes se întoarce codul 201 sau 204 (NO CONTENT) când metoda nu s-a realizat;

<sup>2</sup> © <https://lvivcity.com/single-page-app-vs-multi-page-app>

- DELETE – după cum sugerează și numele, cererea cu metoda DELETE are ca scop ștergerea din sistem a unei anumite resurse specificate întorcând codul 204.

## 4.2 Tehnologii Utilizate

În următoarele subcapitole vor fi prezentate tehnologiile folosite, prezentând pentru fiecare dintre acestea detalii despre funcționarea lor, avantajele utilizării lor și motivul pentru care au fost alese în dezvoltarea aplicației.

### 4.2.1 React [7]

Lansată în 2013 și susținută de Meta, React este o bibliotecă populară open source de JavaScript, datorită performanțelor ridicate, grație reprezentării paginii web prin tehnologia Virtual Document Object Model.

Spre deosebire de DOM-ul tradițional ce reîncarcă pagina la fiecare modificare, VDOM identifică schimbările făcute în pagină crescând astfel eficiența de lucru prin minimizarea manipulărilor.

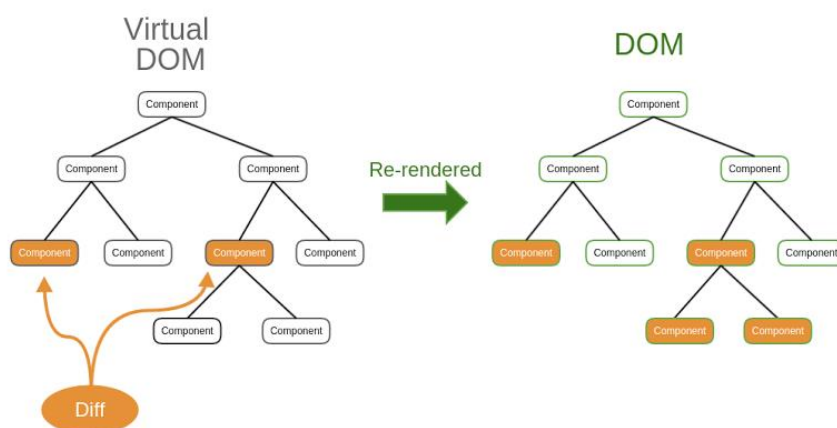


Figura 5: VDOM <sup>3</sup>

Având o arhitectură bazată pe componente, React folosește pentru fiecare dintre acestea o stare, utilizată pentru a stoca date, jucând de asemenea un rol important în logica interacțiunii lor. Mai mult decât atât, cursul datelor este de tip unidirecțional, părinții unei componente transmițându-i date prin „props” imutabili (comparabil cu parametrii unei funcții din limbajul C), copiii la rândul lor comunicând înapoi prin „callbacks”.

Folosind atât TypeScript, în combinație cu sintaxa JavaScript XML, această bibliotecă pune la dispoziție utilizatorilor săi o gamă largă de funcții, componente și tehnici ce

<sup>3</sup> © <https://jayeshinde.medium.com/need-for-virtual-dom-19ebf6843d95>



simplifică masiv procesul de dezvoltare, precum hooks (funcții reutilizabile ce pot controla starea componentelor), React Router, React Native, etc.

### 4.2.2 TypeScript [8]

TypeScript este un limbaj de programare static tipizat, dezvoltat de Microsoft pentru a depăși limitările JavaScript, iar împreună cu React cele două formează o combinație puternică pentru dezvoltarea aplicațiilor web.

Fiind totodată orintat pe obiect, acest limbaj oferă numeroase beneficii ce vin în ajutorul programatorilor, printre care se numără: tipizarea statică ce face ca detecția erorilor să se realizeze într-un mod mai rapid și mai facil, utilizarea modulelor ce duce la structurarea mai bună a codului, aducând lizibilitate, sau implementarea interfețelor garantând funcționalități pentru o anumită clasă sau un obiect și ușurând munca prin crearea de cod reutilizabil.

### 4.2.3 .NET [9]

O platformă de dezvoltare modernă, open source, aleasă timp de 3 ani la rând cel mai îndragit framework de către utilizatorii Stack Overflow, .NET împreună cu limbajul de programare C# reprezintă nivelul logic al acestei aplicații.

Principalele motive care au stat la baza alegerii acestei tehnologii au fost numeroasele beneficii aduse de aceasta, printre care se numără:

- Performanțe crescute – în cazul lucrului cu baza de date sau a serializării JSON-urilor pentru autentificare, .NET excelează din punct de vedere al performanțelor, depășind detașat alte tehnologii similare precum NodeJS sau Java Servlet;

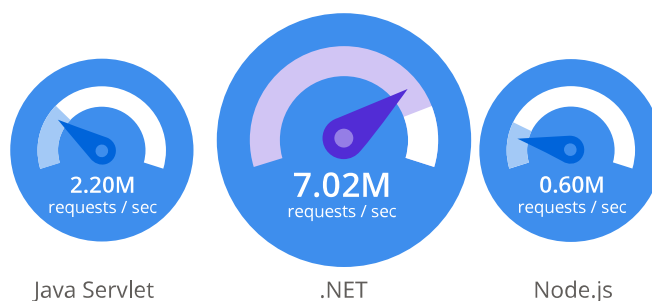


Figura 6: TechEmpower benchmark <sup>4</sup>

---

<sup>4</sup> © <https://dotnet.microsoft.com/en-us/platform/why-choose-dotnet>

- Securitatea – cu suportul Microsoft, acest framework cu capacități de securitate avansate, echipa din spatele lui inovând constant prin actualizări frecvente, oferă funcționalități precum strong typing, autentificări securizate sau CAS, ce întăresc protecția sistemului.

Date fiind inexperiența în dezvoltarea aplicațiilor web, familiaritatea cu limbajele C/C++, dar totodată și dorința de a învăța, un alt motiv care a determinat alegerea acestei platforme de dezvoltare a fost numărul mare de dezvoltatori ce lucrează cu aceasta, existând o documentație extensivă și o serie de forumuri pe baza cărora s-au putut aduna informații legate de procesul de implementare.

#### 4.2.4 PostgreSQL [10]

PostgreSQL este un sistem open-source de gestiune al bazelor de date relaționale, dezvoltat în 1986 la Universitatea din California, Berkeley sub îndrumarea profesorului Michael Stonebreaker.

Principalul motiv al utilizării acestei tehnologii în procesul de dezvoltare îl reprezintă capacitatea de a lucra cu baze de date relaționale. Acestea aduc o multitudine de avantaje în implementarea aplicației, precum structurarea datelor în tabele, utilizarea relațiilor între tabele folosind chei, folosirea limbajului SQL, fapt ce duce la manipularea mai eficientă a informațiilor din sistem și aduce integritate logicii programului.

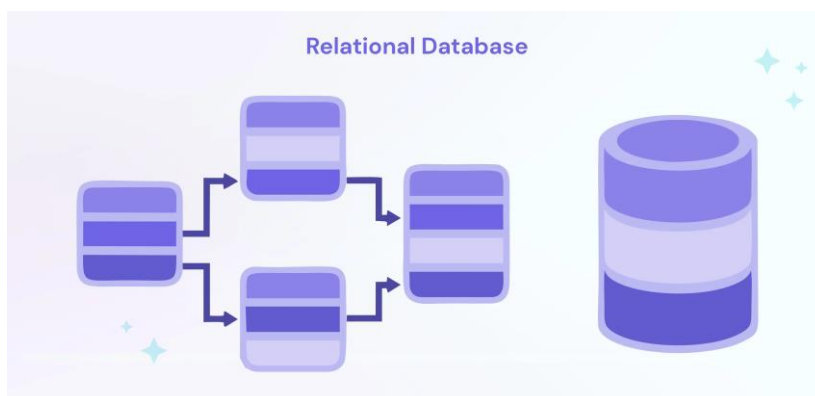


Figura 7: Bază de date relațională <sup>5</sup>

Mai mult decât atât, prin utilizarea unelei DBeaver, o unealtă pentru interacțiunea cu baza de date, cât și cu ajutorul tehnologiei Fluent API, implementarea datelor sistemului s-a

---

<sup>5</sup> © <https://www.boltic.io/blog/relational-databases-vs-nosql>

realizat cu ușurință realizând migrarea de început „InitialCreate”, iar apoi modificând tabelele și relațiile cu migrări adiționale, în funcție de caz.

#### **4.2.5 Docker [11]**

Docker este o platformă open-source, dezvoltată cu scopul de a împacheta aplicații sau procese în „containere”. Unul dintre avantajele cele mai mari ale utilizării acestei tehnologii o reprezintă portabilitatea, aplicația fiind gândită să poată fi utilizată indiferent de sistemul de operare al utilizatorului. Nu în ultimul rând, Docker face ca și containerele create să fie izolate de procesele sistemului gazdă, fapt ce aduce încă un strat de siguranță proiectului.

### **4.3 Diagrama Bazei de Date**

Poate una dintre cele mai importante părți ale unei aplicații moderne ce folosește lucrul cu datele este baza de date, întrucât acesta este spațiul de stocare al informațiilor cu care sistemul lucrează. Totodată aceasta permite o organizare eficientă a datelor, datorită tipului relațional a acesteia și un acces rapid și eficient la date.

Baza de date a acestei aplicații, implementată cu PostgreSQL, este împărțită în 11 tabele cu atribuții variate, între care există relații atât de tip one-to-many, cât și many-to-many.

Pentru o manipulare mai eficientă a datelor și pentru asigurarea integrității acestora relațiile de tip many-to-many au fost implementate folosind tabele de join care fac legătura între două tabele.

O astfel de tabelă are un ID propriu, dar reține și ID-urile tabelelor pentru care face legătura acestea devenind chei străine (foreign key) în proces. ID-urile tabelelor între care se realizează relația many-to-many devin chei primare și astfel această relație se transformă în două de tip one-to-many.

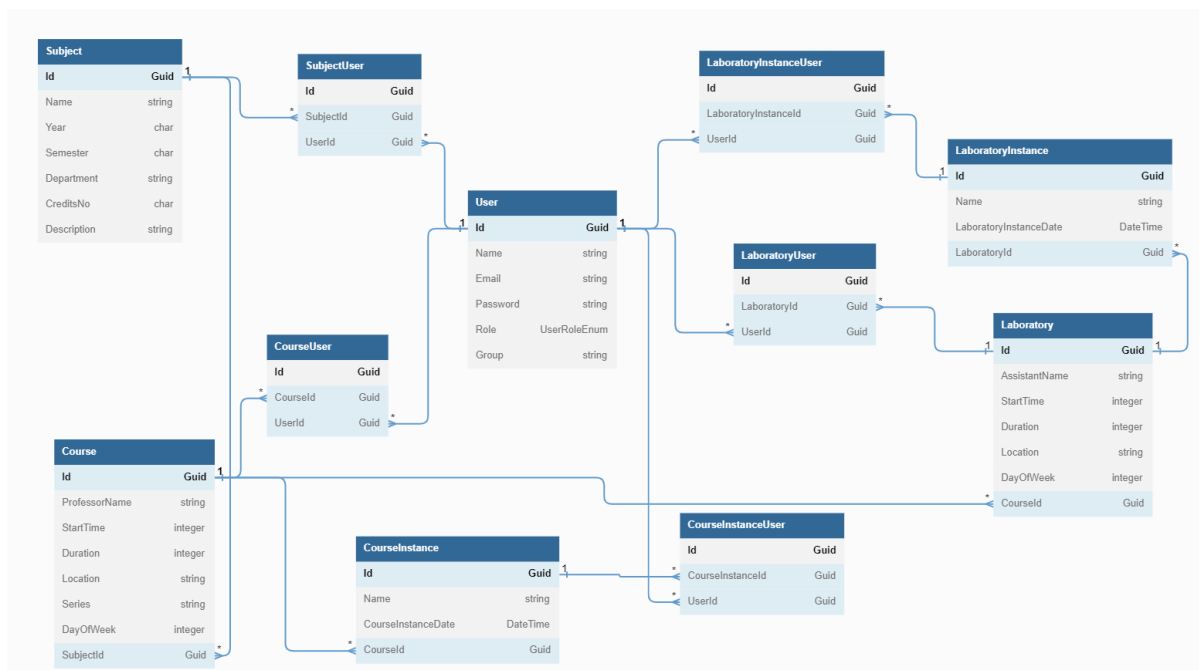


Figura 8: Diagrama bazei de date

În cele ce urmează o să fie explicat rolul fiecărei table din structura bazei de date, explicitând cele mai importante câmpuri și o să se analizeze relațiile ce se leagă între acestea.

User	
Id	Guid
Name	string
Email	string
Password	string
Role	UserRoleEnum
Group	string

Figura 9: Tabela User

Posibil cea mai importantă tabelă dintre toate, întrucât fără ea aplicația nu ar funcționa, User definește câmpuri legate de datele de autentificare ale utilizatorului (email, parolă), dar și informații personale (nume, grupă). Rolul utilizatorului are 3 forme posibile și anume student, cadru didactic („personnel”) sau administrator („admin”), însă acestea vor fi explicate mai amănunțit în subcapitolul următor.

Nu în ultimul rând, în cazul în care utilizatorul are rolul de student această tabelă extinde relații de tip many-to-many cu Subject, Course, CourseInstances, Laboratory, LaboratoryInstances, întrucât s-a mers pe structura: un student poate avea mai multe materii, cursuri, etc. și o materie, un curs poate avea mai mulți studenți.

Subject	
Id	Guid
Name	string
Year	char
Semester	char
Department	string
CreditsNo	char
Description	string

Figura 10: Tabela Subject

Tabela Subject este cea care reține materiile din cadrul unei facultăți, stocând date importante de poziționare în contextul academic, anume: numele, anul în care materia este predată, numărul de credite aferent, etc.

La nivel de relații între tabele, aceasta are una de tip many-to-many cu Users și una one-to-many cu Subjects, conform logicii: o materie are mai multe cursuri în funcție de serie.

Course	
Id	Guid
ProfessorName	string
StartTime	integer
Duration	integer
Location	string
Series	string
DayOfWeek	integer
SubjectId	Guid

Figura 11: Tablea Course

Tabela Course reprezintă cursul unei materii susținute de un anumit profesor și stochează date relevante pentru acesta. Astfel, în câmpurile tabelului sunt stocate informații precum: numele profesorului, seria, locația sau ziua în care se ține cursul (1 – luni, 2 – marți, etc.) fiind reținute sub formă de „integer” pentru simplitate în ordonarea acestora pentru afișarea orarului.

Relațiile în care este asociată Course sunt many-to-many cu Users, one-to-many cu CourseInstances (un curs are mai multe instanțe) și one-to-many cu Laboratory (un anumit curs are mai multe laboratoare).

CourseInstance	
<b>Id</b>	<b>Guid</b>
Name	string
CourseInstanceDate	DateTime
Courseld	Guid

Figura 12: Tabela Course Instance

Această tabelă desmnează o instanță a unui anumit curs de la o anumită dată, fiind reținute numele acestuia, data și ID-ul cursului de care aparține. Course Instance se află în relație many-to-many cu User și many-to-one cu Subject așa cum a fost explicat anterior.

Laboratory	
<b>Id</b>	<b>Guid</b>
AssistantName	string
StartTime	integer
Duration	integer
Location	string
DayOfWeek	integer
Courseld	Guid

Figura 13: Tabela Laboratory

Tabela Laboratory reține datele unui laborator și are câmpuri definitorii pentru acesta cum ar fi: numele asistentului care îl predă, locația unde se ține, ziua sau ID-ul cursului de care aparține. Laboratory are relații de tip many-to-many cu tabela User, many-to-one cu Subject și one-to-many cu LaboratoryInstance (un laborator are mai multe instanțe).

LaboratoryInstance	
<b>Id</b>	<b>Guid</b>
Name	string
LaboratoryInstanceDate	DateTime
LaboratoryId	Guid

Figura 14: Tabela Laboratory Instance

Similară cu Course Instance, tabela Laboratory Instance are ca rol definirea unei anumite instanțe de laborator, reținând totodată numele acesteia, data și ID-ul laboratorului de proveniență. Aceasta se află în relație de many-to-many cu Users și many-to-one cu Laboratory.

Pe lângă aceste tabele, mai există alte 5 care deserveșc rolul de tabele de joncțiune și anume SubjectUser, CourseUser, CourseInstanceUser, LaboratoryUser, LaboratoryInstanceUser. Așa cum a fost explicat și la începutul subcapitolului, acestea au rolul de a eficientiza manipularea datelor, iar în cele ce urmează vom explica mai detaliat modul în care funcționează doar o singură astfel de structură, pentru evitarea redundanței.

SubjectUser	
<b>Id</b>	<b>Guid</b>
SubjectId	Guid
UserId	Guid

Figura 15: Tabela Subject User

În cazul tabelii Subject User, există 2 relații și anume one-to-many între Subject și SubjectUser și one-to-many între SubjectUser și User. Așadar, pentru prima relație enunțată,

câmpul ID al tabelii Subject joacă rolul de cheie primară, iar SubjectId din tabela SubjectUser devine cheie străină. Celelalte relații și tabele de acest tip sunt realizate în mod similar, cu mențiunea că se folosesc alte tabele.

#### 4.4 Tipuri de Utilizatori

După analiza cerințelor, s-a descoperit nevoia ca fiecărui utilizator din cadrul aplicației, să îi fie acordat un rol în funcție de atribuțiile sale, acest lucru întâmplându-se în momentul în care acesta este înregistrat. Considerând nevoile fiecăruia dintre aceștia, avem astfel trei tipuri de utilizatori:

- Student – deși în cadrul aplicației este cel mai întâlnit tip de utilizator acesta are și cele mai puține drepturi dintre cele trei tipuri de roluri. Studentul are acces la orarul său, în funcție de laboratoarele și cursurile la care a fost înrolat, poate să se înregistreze la o oră la care a participat și să își vadă prezențele pe care le-a acumulat pe parcursul semestrului;
- Personnel (cadru didactic) – acest rol are mai multe drepturi decât cel anterior menționat, întrucât are acces la panoul materiilor sale, de unde poate crea o instanță de curs sau de laborator, iar din pagina creată, studenții pot scana codul QR pentru înregistrare. La fel ca și studenții, cadrele didactice pot de asemenea să își vizioneze orele pe care trebuie să le predea sau să își vadă propriile cursuri și laboratoare susținut anterior;
- Admin (administrator) – cel mai important rol, fiind și cel care are și cele mai multe drepturi, administratorul joacă un rol esențial în procesul aplicației întrucât este cel care este responsabil de configurarea și gestionarea acesteia. Administratorul are îndatoriri precum crearea materiilor, a cursurilor și a laboratoarelor, gestionarea utilizatorilor, precum și modificarea unor parametri ai sistemului, având totodată acces la toate datele bazei de date.



## 5 DETALII DE IMPLEMENTARE

În următoarele pagini ale lucrării vor fi prezentate atât pașii care au fost urmați în inițializarea proiectului, cât și funcționalitățile pe care soluția le implementează în conformitate cu cerințele utilizatorilor.

### 5.1 Inițializarea Proiectului

Așa cum a fost prezentat și la punctul 4.2, pentru realizarea proiectului au fost alese o serie de tehnologii, selectate împreună pentru a da un rezultat cât mai bun și care trebuie să fie legate pentru a comunica.

Astfel, la începutul proiectului s-a creat un container de Docker pentru ca implementarea și conexiunea cu baza de date, integrată cu PostgreSQL, să fie similar cu cele ale aplicațiilor moderne.

După ce toate configurările necesare au fost realizate la pasul anterior, s-a inițializat proiectul în partea de backend (.NET) creându-se fișierul „appsettings.json” și specificând căile necesare pentru realizarea conexiunii.

```
1 {
2   "AllowedHosts": "*",
3   "CorsConfiguration": {
4     "Origins": [
5       "http://localhost:5000",
6       "http://localhost:3000"
7     ]
8   },
9   "JwtConfiguration": {
10    "Key": "SecretKey000000000000000000",
11    "Issuer": "https://my.app",
12    "Audience": "https://my.app"
13  },
14  "ConnectionStrings": { // here usually all database connection strings are set.
15    "WebAppDatabase": "Server=localhost;Port=5432;Database=PoliHub;User Id=PoliHub;Password=PoliHub;"
16  },
17  "FileStorageConfiguration": {
18    "SavePath": "../TempDocuments" // this path is where the file service saves the files on the filesystem
19  }
20 }
21
```

Figura 16: Fișierul appsettings.json

După cum se poate observa și în imagine, aplicația a fost realizată local, căile de acces pentru întărirea comunicării fiind:

- Back-end: <http://localhost:5000>
- Front-end: <http://localhost:3000>
- Baza de date: <http://localhost:5432>

Motivul integrării locale al aplicației îl reprezintă gradul ridicat de securitate și limitarea studenților de a se conecta de pe alte rețele, decât cea în cauză, fapt ce sporește precizia datelor de prezență ale acestora.

Aplicația fiind una de tip RESTful API și întrucât dezvoltarea acesteia a fost realizată prin implementarea back-end-ului prima oară, s-a ales Swagger [12] atât pentru testarea endpoint-urilor, cât și pentru a oferi o privire de ansamblu mai bună asupra proiectului.

În partea de front-end a proiectului, s-a folosit Vite [13] (fișierul de configurare numit „.env” fiind disponibil în cadrul anexei 1) pentru a servi aplicația ca un server local și pentru a încapsula codul în vederea optimizării timpilor de rulare. Vite ajută de asemenea și la procesul de dezvoltare întrucât, în urma schimbărilor din cod, acesta afișează rapid pagina aplicației înlocuind doar modificările făcute.

Configurarea de rețea a server-ului Vite este prezentată în cadrul anexei 2, în care este arătat un script de configurare, prin care se extrage adresa IP a PC-ului gazdă, care se modifică în fișierul „.env”, acesta trebuind să fie rulat în momentul în care computer-ul sau rețeaua de pe care funcționează aplicația sunt schimbate.

Observație: întrucât Swagger și Vite nu joacă un rol major în dezvoltarea aplicației acestea au fost introduse în această secțiune pentru a aduce claritate lucrării.

## 5.2 Funcționalități

După cum a fost enunțat și în capitolul 2, aplicația a fost realizată, în primul rând, cu interesul utilizatorilor în minte, adunând astfel o suită de cerințe de natură atât funcțională, cât și nefuncțională. Așadar, în următoarele subcapitole se va descrie modul în care cerințele au fost întrunite, realizând totodată o prezentare amănunțită a interfețelor grafice și prezentând modul lor de funcționare.

### 5.2.1 Operații CRUD

Principalele operații necesare pentru lucrul cu datele sunt operațiile: „Create” „Read” „Update” și „Delete”. Așa cum a fost explicat și în subcapitolul 4.1, nivelul de prezentare al aplicației comunică cu backend-ul prin endpoint-urile expuse de acesta.

În cadrul aplicației, în funcție de tabelele pe care se acționează, sunt prezente endpoint-urile pentru User, Subject, Course, CourseInstance, Laboratory, LaboratoryInstance (reprezentate de „[x]”):

- GET/api/[x]/GetById/{id}
- GET/api/[x]/GetPage
- POST/api/[x]/Add
- PUT/api/[x]/Update
- DELETE/api/[x]/Delete/{id}

Pentru tabelele Subject, Course, CourseInstance, Laboratory, LaboratoryInstance (reprezentate de „[x]”) s-au implementat adițional pentru ștergerea și adăugarea utilizatorilor și:

- POST/api/[x]/AddUserTo[x]
- DELETE/api/[x]/Delete/DeleteUserFrom[x]{id}

Pentru tabela User au fost realizate și endpoint-uri pentru vizualizarea orarului și a prezențelor la ore:

- GET/api/User/GetTimetableClasses/{id}
- GET/api/User/GetAttendances/{id}

De asemenea, pentru autentificarea utilizatorilor s-au creat:

- POST/api/Authorization/Login
- POST/api/Authorization/Register

### 5.2.2 Bara de Navigare

Pentru un acces mai facil la toate funcționalitățile aplicației, utilizatorii au la dispoziție în partea de sus a paginilor o bară de navigare, cu diferite link-uri către părți ale site-ului în funcție de rolul individului.

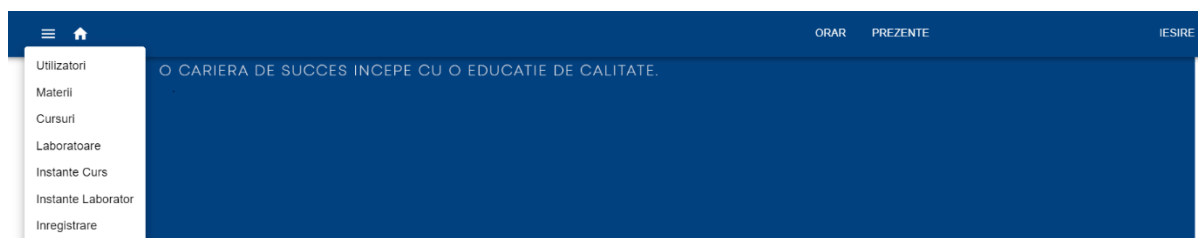


Figura 17: Bara de navigare

Pentru utilizatorii obișnuiți (studenți și cadre didactice), aceștia au opțiunea de a accesa pagina principală (pictograma în formă de casă), orarul sau prezențele lor, iar în partea din dreapta aceștia au opțiunea de a ieși din cont sau de a se autentifica, după caz.

Adițional, administratorul sistemului are un meniu în partea din stânga a bării de navigare, prin care poate accesa baza de date în funcție de tabelele pe care dorește să le vizualizeze sau modifice, acelea fiind paginile de: utilizatori, materii, cursuri, laboratoare, instanțe de curs, instanțe de laborator sau înregistrarea unui cont nou.

### 5.2.3 Autentificare

În primul rând, pentru a accesa funcționalitățile portalului web, utilizatorii aceștia trebuie să se autentifice. Atât studenții, cât și profesorii vor fi înregistrați de către

administratorul sistemului, acesta având la dispoziție o pagină de înregistrare în mijlocul căreia se află un formular în care trebuie să completeze datele utilizatorilor.

Inregistrare

Nume \*

Va rugam sa introduceti Nume

E-mail \*

Va rugam sa introduceti E-mail

Parola \*

Va rugam sa introduceti Parola

Rol \*

Va rugam sa introduceti Rol

Grupa \*

Va rugam sa introduceti Grupa

TRIMITE

Aveti deja un cont? [Autentificare](#)

Figura 18: Pagina de înregistrare

Câmpurile ce trebuie introduse sunt numele, e-mail-ul, parola, rolul și grupa din care face parte utilizatorul, cu precizările că rolul poate fi doar de 2 tipuri: „Student” (pentru student) sau „Personnel” (pentru personalul didactic), iar grupa în cazul profesorilor/asistenților va avea valoarea „-”, întrucât aceasta nu va fi utilizată.

Autentificare

E-mail \*

admin@default.com

Parola \*

admin@default.com

Gestionează parolele...

TRIMITE

Figura 19: Pagina de autentificare

După ce administratorul înregistrează utilizatorii, acesta le va trimite un e-mail în care le specifică parola pe care o vor utiliza pentru autentificare. Aceștia vor își introduce e-mail-ul și parola și vor fi redirecționați la pagina principală a aplicației, în cazul în care datele introduse sunt valide. În caz contrar, utilizatorul va primi un mesaj de eroare, fiind specificat faptul ca datele introduse nu sunt corecte.

În figura 18 se poate observa mecanismul de autocompletare a datelor, câmpurile salvând informația de autentificare în cazul în care utilizatorul a mai accesat platforma în trecut, aspect care s-a realizat cu ajutorul utilizării tehnologiei de „cookies”. În caz contrar, câmpurile vor fi goale.

#### 5.2.4 Pagina Principală

Așa cum a fost explicat anterior, după autentificare persoana ce folosește aplicația va fi trimisă la pagina principală, unde poate accesa diferite resurse atât interne, cât și externe site-ului.



Figura 20: Pagina principală

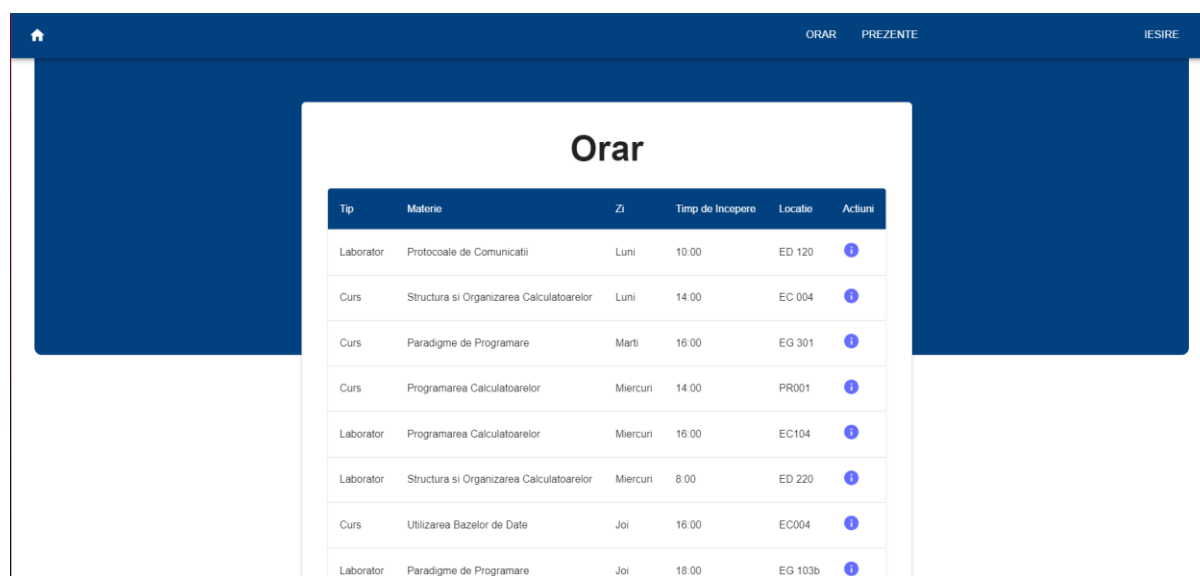
Aceasta este caracterizată de un design minimalist și intuitiv prin care studenții și profesorii pot accesa o serie de 5 resurse:

- Orar – cursurile și laboratoarele la care utilizatorii trebuie să participe;
- Prezențe – înregistrările precedente la cursuri și laboratoare ale studenților;
- upb.ro - site-ul oficial Universitatea Politehnica București ([www.upb.ro](http://www.upb.ro));
- studenti.pub - site-ul administrativ unde studenții își pot vedea situația școlară și alte informații adiționale legate de parcursul lor universitar ([www.studenti.pub.ro](http://www.studenti.pub.ro)) ;

- Moodle - portalul de e-learning cu diferite funcționalități didactice precum: încărcarea temelor, accesarea materialelor de curs, vizualizarea punctajului parțial la o anumită materie, etc. (<https://curs.upb.ro/2022/>).

#### 5.2.5 Vizualizarea Orarului

Modalitatea prin care atât studenții, cât și profesorii pot vizualiza cursurile și laboratoarele la care trebuie să participe se realizează prin accesarea paginii „orar” din bara de navigare sau din pagina principală.



Tip	Materie	Zi	Timp de Incepere	Locatie	Actiuni
Laborator	Protocoale de Comunicatii	Luni	10.00	ED 120	
Curs	Structura si Organizarea Calculatoarelor	Luni	14.00	EC 004	
Curs	Paradigme de Programare	Marti	16.00	EG 301	
Curs	Programarea Calculatoarelor	Miercuri	14.00	PR001	
Laborator	Programarea Calculatoarelor	Miercuri	16.00	EC104	
Laborator	Structura si Organizarea Calculatoarelor	Miercuri	8.00	ED 220	
Curs	Utilizarea Bazelor de Date	Joi	16.00	EC004	
Laborator	Paradigme de Programare	Joi	18.00	EG 103b	

Figura 21: Pagina Orar

Orele sunt afișate sub forma unui tabel cu intrări ce reprezintă un curs sau un laborator, ordonate crescător după ziua și timpul de începere și are ca și coloane: tipul orei (laborator/curs), materia de care aparține, ziua în care se ține, timpul de începere și locația.

De asemenea, în partea din dreapta, ultima coloană intitulată „acțiuni” are rolul de a trimite utilizatorul la pagina laboratorului sau cursului respectiv, o funcție importantă atât pentru profesori cât și pentru studenți, întrucât cei din urmă accesează pagina respectivă pentru a crea o nouă instanță la acea materie.

#### 5.2.6 Vizualizarea Prezențelor

Folosind pagina de prezențe, utilizatorii își pot vedea cursurile, laboratoarele sau seminariile la care au participat în decursul anului universitar.



Prezente Cursuri			
Nume	Materie	Data Curs	Actiuni
Curs 1 - Introducere	Programarea Calculatoarelor	18/5/2023	<a href="#">i</a>
Cursul 0. Intro. Legea lui Moore. Istoria sistemelor de calcul	Structura si Organizarea Calculatoarelor	24/6/2023	<a href="#">i</a>
Cursul 2 - Analiza Soft	Programarea Calculatoarelor	29/5/2023	<a href="#">i</a>

Prezente Laboratoare/Seminarii			
Nume	Materie	Data Laborator	Actiuni
Laborator 2 - Nivelul de Transport	Protocoale de Comunicatii	20/5/2023	<a href="#">i</a>
Laborator 1 - Introducere	Programarea Calculatoarelor	18/5/2023	<a href="#">i</a>

Figura 22: Pagina Prezențe

Această pagina conține două tabele: „prezențe cursuri” și „prezențe laboratoare/seminarii”, ambele afișând numele orei, materia și data acesteia. Adicional, profesorii vor putea accesa pagina instanței respective, pentru a vizualiza studenții care au fost anterior prezenți.

#### 5.2.7 Vizualizarea Materiilor, Cursurilor și a Laboratoarelor

Modalitatea prin care se poate vizualiza o materie, un curs sau un laborator din cadrul facultății se face accesând pagina respectivă, care îi poartă și numele. Pentru evitarea redundanței, s-a ales prezentarea unei singure astfel de pagini și anume pagina materie, întrucât restul paginilor menționate au un conținut similar, cu modificări minimale ce vor fi prezentate în cele ce urmează.

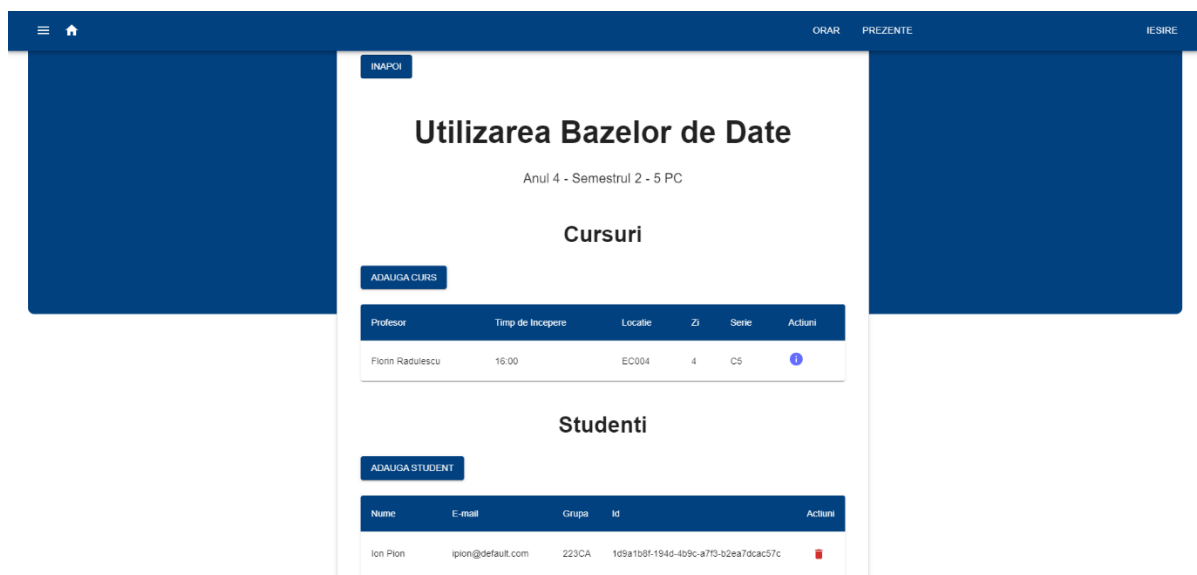


Figura 23: Pagina Materie

Astfel, în exemplul acestui tip de pagină se pot observa datele specifice ale disciplinei (nume, an, semestru, numărul de puncte credit valorate), dar și o serie de butoane și tabele.

Cele două tabele prezente în pagină au rolul de a afișa cursurile și studenții materiei respective. În primul tabel, acesta expune numele titularului de curs, ora de început, ziua și seria cursului, având posibilitatea ca utilizatorul să fie transportat la pagina cursului respectiv prin apăsarea pictogramei din ultima coloană și anume cea de acțiuni. În cazul tabelului de studenți le sunt afișate datele acestora de identificare: nume, e-mail, grupă și ID.

În cazul în care utilizatorul este un administrator, acesta va avea acces la trei butoane ce îi oferă posibilitatea de a adăuga un curs, înrola un student în cadrul materiei sau de a elimina un student de la disciplină.

Dacă butonul „adaugă student” este apăsător, se va genera un element de tip „pop-up”, în care va fi cerut ID-ul studentului pe care se dorește să fie înrolat în cadrul materiei. Prin apăsarea butonului „adaugă curs” administratorul este întâmpinat de un formular în care trebuie să introducă datele cursului respectiv, ID-ul materiei fiind cel specific materiei de pe pagina respectivă și este completat automat.



### Adauga curs

Profesor *	Timp de Incepere *
<input type="text" value="Va rugam sa introduceti Profeso"/>	<input type="text" value="Va rugam sa introduceti Timp de"/>
Durata (min.) *	Serie *
<input type="text" value="0"/>	<input type="text" value="Va rugam sa introduceti Serie"/>
Locatie *	Zi *
<input type="text" value="Va rugam sa introduceti Locatie"/>	<input type="text" value="0"/>
Id Materie *	
<input type="text" value="179cd958-bb57-4563-8991-c2et"/>	

TRIMITE

Figura 24: Formular pentru adăugarea unui curs

În cazul paginilor de curs sau de laborator, profesorii sau administratorii vor putea crea în loc de cursuri, instanțe de curs sau de laborator, afișându-se un formular pentru tipul de oră în cauză, în care aceștia vor trebui să completeze datele necesare.

#### 5.2.8 Înregistrarea Prezențelor

După ce pagina instanței de curs sau de laborator a fost creată, profesorii o vor accesa așa cum a fost prezentat anterior (din orar) și o vor afișa pe ecranul computer-ului studenților prezenți.

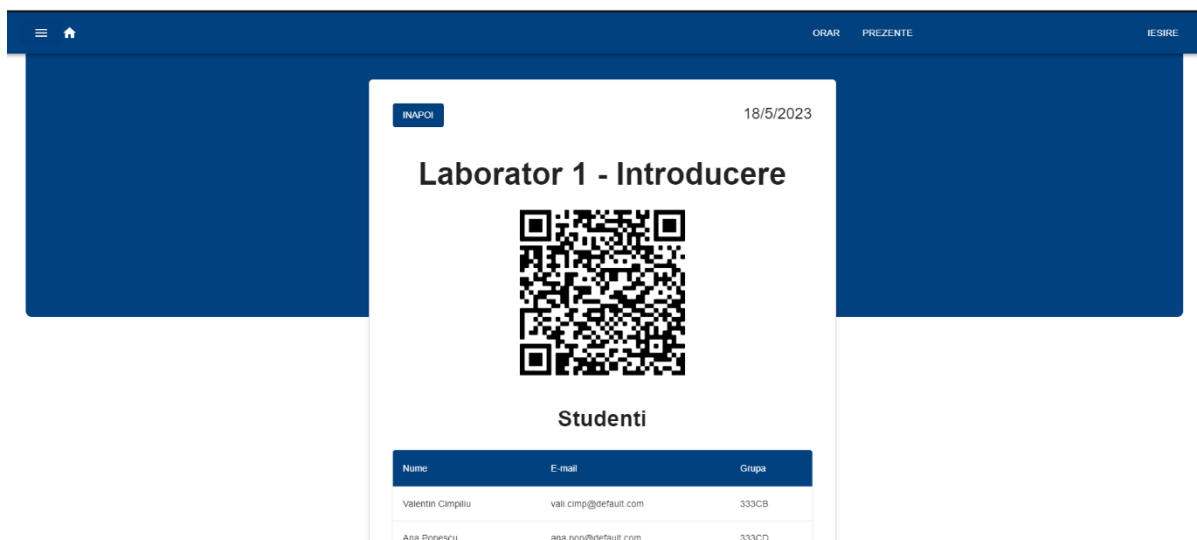


Figura 25: Pagina Instanță de Laborator

Codul QR creat cu componenta „QRCode” din cadrul bibliotecii „react-qr-code” este generat prin codificarea URL-ului paginii curente, în cazul figurii 25 el fiind „http://localhost:3000/laboratoryInstances/078c7e20-dcb8-431d-a2f9-bbab9c25cf12”. Pentru a permite conectarea studenților însă, în loc de „localhost” este trecută adresa IPv4 de LAN a PC-ului de pe care rulează aplicația, ea reprezentând modalitatea de adresare a dispozitivelor studenților cu computer-ul profesorului. Această adresă este preluată din cadrul fișierului de configurare a variabilelor de mediu (disponibil în cadrul anexei 2) pe care pornește server-ul de front-end.

După ce profesorul afișează pagina studenților, aceștia prin scanarea codului QR cu ajutorul camerei telefonului mobil propriu, vor fi redirecționați la pagina mai sus menționată (figura 25), primind un mesaj de confirmare ce demonstrează că au fost înregistrați ca fiind prezenți. În cazul în care aceștia nu sunt autentificați, vor fi redirecționați la pagina de login prima oară, apoi după introducerea datelor vor fi catalogați ca fiind prezenți.

#### 5.2.9 Gestionarea Datelor de către Administrator

Pentru gestionarea mai facilă a datelor de către administratorul de sistem s-a implementat un meniu adițional pentru acesta, prin care să poată avea acces la toate datele din sistem, fiind prezentată doar una dintre ele și anume pagina de materii.

Materii							
<div> <div>ADAUGA MATERIE</div> <div></div> </div>							
<div>Intrari pe pagina 10 1-8 din 8</div>							
Nume	An	Semestru	Departament	Puncte Credit	Descriere	Actiuni	
Structura si Organizarea Calculatoarelor	2	2	SD	5	SOC		
Protocoale de Comunicatii	2	2	PCM	6	PCom		
Modelare si Simulare	2	2	RT	5	MS		
Paradigme de Programare	2	2	LLT	5	PP		
Utilizarea Bazelor de Date	4	2	BD	5	<a href="https://curs.upb.ro/2022/course/view.php?id=5114">https://curs.upb.ro/2022/course/view.php?id=5114</a>		
Arhitectura Sistemelor de Calcul	3	1	SMC	5	ASC		

Figura 26: Pagina Materii

Din această pagină, administratorul sistemului poate vizualiza materiile având și o bară de căutare prin care poate specifica numele (de exemplu) unei materii, datele fiind paginate și putând fi schimbate numărul intrărilor afișate. Butonul „adaugă materie” afișează un „pop-up” cu un formular pentru introducerea datelor, însă dacă se dorește modificarea unei intrări, se poate realiza acest lucru prin dublu-click pe materia respectivă. De asemenea există și posibilitatea de a șterge o disciplină, sau a vizualiza pagina sa, prin apăsarea butonului respectiv din coloana de acțiuni.

### 5.3 Aspecte de Securitate

Un punct vital al oricărei aplicații web este reprezentat de securitatea sa, atacurile cibernetice putând cauza efecte devastatoare atât asupra datelor utilizatorilor acesteia, prin periclitarea integrității personale, cât și asupra sistemului în sine.

În decursul proiectării și dezvoltării proiectului s-a pus un accent deosebit pe protecția acestuia, implementând o serie de măsuri de apărare împotriva atacurilor precum: autentificarea cu token-uri JSON, stocarea encodată a datelor, sau păstrarea sistemului în rețeaua locală.

JSON Web Token (JWT) [14] reprezintă un standard RFC 7519 ce definește un obiect de tip JSON, în care sunt încapsulate date cu privire la informațiile unui anumit utilizator, semnat cu o semnătură digitală și trimise într-o manieră securizată.

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1aWw1laWQoIi4YTA4ZTM0NC1jNjc2LTRjNDItOGRkMi00YmE4ZDI3Zjk2ZjIiLCJuYW1lIjoiaWRtaW4iLCJlbWFnZWVudCI6ImFkbWluQGRLZmF1bHQuY29tIiwicm9sZSI6eyJOYW1lIjoiaWRtaW4iLCJwYXN1ZSI6IkFkbWluIn0sIm5iZiI6MTY4NzYzMzAwNmMzImVycyI6Imh0dHBzOi8vbXBuIiwiaWF0IjoiYXVkaHR0eHM6Ly9teS5hcHAifQ.ubk6QVn7XoU8tftN0ZF-tu1X3BKC1xuT_JiYA5Qk2Yo
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYLOAD: DATA

```
{  
  "nameid": "8a08e344-c676-4c42-8ad2-4ba8d27f96f2",  
  "name": "Admin",  
  "email": "admin@default.com",  
  "role": {  
    "Name": "Admin",  
    "Value": "Admin"  
  },  
  "nbf": 1687630073,  
  "exp": 1688234873,  
  "iat": 1687630073,  
  "iss": "https://my.app",  
  "aud": "https://my.app"  
}
```

În figura de mai sus este prezentată anatomia unui astfel de token, atât în formă codificată cât și decodificată. În antetul JWT-ului este specificat tipul algoritmului de criptare și anume HS256 (HMAC SHA256) [15], ce generează o criptare (aproape) unică, dat fiind un astfel de token.

## 6 STUDIU DE CAZ / EVALUAREA REZULTATELOR

Folosind cerințele utilizatorilor descrise în cadrul capitolului 2 : „Analiza și Specificarea Cerințelor” s-a explicat implementarea lor în întregime în partea 5.2 de detaliere a funcționalităților aplicație, atingând toate calitățile cerute de client. După testarea funcționalităților, aplicația a fost prezentată unui grup de studenți ce s-au declarat mulțumiți după interacțiunea cu aceasta.

Design-ul modern, minimalist și intuitiv a produs o experiență de navigare facilă în cadrul portalului web, utilizatorii putând să ajungă cu ușurință la informațiile pe care doreau să le acceseze. Mai mult decât atât, persoanele ce au folosit aplicația au apreciat performanțele modelului, întrucât au putut să beneficieze eficient de funcțiile acestuia, datorită timpului scurt de răspuns.

Cu toate acestea, este importantă recunoașterea limitărilor actuale ale sistemul dorind să se aducă îmbunătățiri ulterioare modelului. Printre posibilele adăugări în vederea evoluției aplicației se numără: integrarea cu platforma Moodle, pentru o experiență pedagogică mai completă sau implementarea unor straturi adiționale pentru combaterea abuzurilor, precum folosirea de date biometrice pentru autentificare sau a cardurilor RFID, comparativ cu cele ce au fost descrise în capitolul 3.

Așadar, evaluările din punct de vedere al eficacității, securității, robusteții și portabilității alături de feedback-ul pozitiv al utilizatorii țintă și satisfacția acestora în legătură cu produsul finit au confirmat ipotezele de la care s-a pornit în construirea proiectului.

## 7 CONCLUZII

Domeniul educației ar trebui să reprezinte o preocupare pentru noi toți, fiind esențial pentru viitorul unei societăți, în care resursa umană să fie bine instruită, performantă și eficientă. În contextul lumii moderne, într-o continuă dezvoltare tehnologică, nici sectorul educației nu este imun la astfel de schimbări, astfel „PoliHub” dorind să aducă o eficientizare în cadrul mediului academic.

Soluția propusă, definește un sistem de monitorizare a prezențelor, menit să reducă timpul consumat pentru activități administrative pentru ca eforturile profesorilor să fie îndreptate doar către activitățile pedagogice.

Funcțiile aplicației propuse împiedică încercările de fraudare ale studenților, grație accentului pus pe securitate și de asemenea, utilizării unui model ce lucrează pe o rețea locală, ameliorând astfel erorile cauzate de factorul uman.

Digitalizarea continuă a demonstrat nevoia utilizării unor tehnologii moderne, care să îndeplinească cerințele utilizatorilor. În acest sens, lucrarea demonstrează o abordare scalabilă, robustă și intuitivă ce depășește limitările actuale ale modelului tradițional de management al prezențelor, incluzând totodată și o manieră intuitivă prin care atât studenții, cât și profesorii își pot vizualiza orarul.

Concluzionând, „PoliHub” nu numai că facilitează munca depusă de cadrele didactice, ci oferă și o transparență sporită, îmbunătățind totodată accesul studenților la date, în dorința unei evoluții constante a legăturii dintre învățământ și tehnologie .

## 8 BIBLIOGRAFIE

- [1] A. Varadannanavar, A. Nair, A. Kumar, P. K and A. R. Choudhury, "Automatic Attendance System Using Biometric Authentication and Varying QR Code," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 2425-2429, doi: 10.1109/ICAC3N56670.2022.10074099.
- [2] U. Koppikar, S. Hiremath, A. Shiralkar, A. Rajoor and V. P. Baligar, "IoT based Smart Attendance Monitoring System using RFID," 2019 1st International Conference on Advances in Information Technology (ICAIT), Chikmagalur, India, 2019, pp. 193-197, doi: 10.1109/ICAIT47043.2019.8987263.
- [3] Sumita Nainan, Romin Parekh, Tanvi Shah, "RFID Technology Based Attendance ManagementSystem", IJCSI International Journal of Computer Science Issues, Vol. 10 -January 2013
- [4] „Three-Tier Architecture”, technopedia.com, 2023 . [Online]. Available: <https://www.techopedia.com/definition/24649/three-tier-architecture>. [Accessed June 2023].
- [5] „What is three-tier architecture?”, ibm.com, [Online]. Available: <https://www.ibm.com/topics/three-tier-architecture>. [Accessed June 2023].
- [6] „Available HTTP Methods”, doc.oro-inc.com, [Online]. Available: <https://doc.oro-inc.com/api/http-methods/>. [Accessed June 2023].
- [7] „React A JavaScript library for building user interfaces”, legacy.reactjs.org, [Online]. Available: <https://legacy.reactjs.org>. [Accessed June 2023].
- [8] „TypeScript Documentation”, typescriptlang.org, [Online]. Available: <https://www.typescriptlang.org/docs/>. [Accessed June 2023].
- [9] „What is .NET?”, dotnet.microsoft.com, [Online]. Available: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>. [Accessed June 2023].
- [10] „PostgreSQL 15.3 Documentation”, The PostgreSQL Global Development Group, [Online]. Available: <https://www.postgresql.org/files/documentation/pdf/15/postgresql-15-A4.pdf>. [Accessed June 2023].
- [11] „What is a container?”, docs.docker.com, [Online]. Available: <https://docs.docker.com/get-started/#what-is-a-container>. [Accessed June 2023].
- [12] „What Is Swagger?”, swagger.io, [Online]. Available: <https://swagger.io/docs/specification/2-0/what-is-swagger/>. [Accessed June 2023].
- [13] „Getting Started”, vitejs.dev, [Online]. Available: <https://vitejs.dev/guide/>. [Accessed June 2023].

[14] „Introduction to JSON Web Tokens”, jwt.io, [Online]. Available: <https://jwt.io/introduction>. [Accesed June 2023].

[15] „US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)”, D. Eastlake 3rd, T. Hansen, May 2011, [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc6234>. [Accesed June 2023].



## 9 ANEXE

```
1 VITE_DEFAULT_LOCALE=ro
2 VITE_APP_TITLE=PoLiHub
3 VITE_APP_API_BASE_URL=http://192.168.1.130:5000|
```

Anexa 1: Fișier de configurare a variabilelor de mediu (.env)

```
1 # Get IP addresses
2 $ipAddresses = Get-NetIPAddress | Where-Object {
3     $_.InterfaceAlias -like "Wi-Fi" -and
4     $_.AddressFamily -eq "IPv4" -and
5     $_.PrefixOrigin -eq "Dhcp"
6 }
7
8 # Extract the IPAddress property
9 $ip = $ipAddresses | Select-Object -ExpandProperty IPAddress
10
11 # Get the current directory path
12 $currentDirectory = Split-Path -Parent $MyInvocation.MyCommand.Path
13
14 # Construct the file path to ".env" in the current directory
15 $file = Join-Path -Path $currentDirectory -ChildPath ".env"
16
17 # Read the file content
18 $fileContent = Get-Content $file
19
20 # Replace IP in VITE_APP_API_BASE_URL
21 $newContent = $fileContent -replace '(VITE_APP_API_BASE_URL=http://)([d.]+(:5000))', "`${1}$ip`${2}"
22
23 # Write the updated content back to the file
24 $newContent | Out-File -FilePath $file -Encoding UTF8 -Force
```

Anexa 2: Fișier pentru configurările de rețea (network\_config.ps1)