Sistem informatic pentru colectarea, gestionarea si analiza datelor medicale in chirurgia pediatrică de urgență

Universitatea de Vest din Timișoara – Facultatea de Matematică și Informatică/

UMF “Victor Babeș” Timișoara

Lucaciu Mihai-Remus, Iancu Mara-Ștefania, Manea Roberta-Maria, Leuca Romina-Anamaria

Coordonator – Dr. Conf. Radu Moleriu, Dr. Călin Muntean

CUPRINS

[**PLANIFICAREA** 5](#_Toc199687829)

[SCOP 5](#_Toc199687830)

[CONSTRANGERI DE PROIECTARE 5](#_Toc199687831)

[ARHITECTURA SISTEMULUI 5](#_Toc199687832)

[PROIECTARE DETALIATA – HARDWARE SI SOFTWARE 6](#_Toc199687833)

[PROIECTAREA BAZEI DE DATE 7](#_Toc199687834)

[REFERINTE DE PROIECT 8](#_Toc199687835)

[SECURITATEA SI INTEGRITATEA PROIECTULUI 9](#_Toc199687836)

[**ANALIZA** 10](#_Toc199687837)

[CERINTE DE PROIECT 10](#_Toc199687838)

[*1. Cerințe funcționale:* 10](#_Toc199687839)

[*2. Cerințe nefuncționale:* 10](#_Toc199687840)

[CAZURI DE UTILIZARE 11](#_Toc199687841)

[*1.Login* 11](#_Toc199687842)

[*2.Adaugare pacient* 11](#_Toc199687843)

[*3.Adaugare rezultate laborator* 12](#_Toc199687844)

[*4.Calculare scor clinic* 12](#_Toc199687845)

[*5.Vizualizare date* 13](#_Toc199687846)

[*6.Monitorizare pacienti* 13](#_Toc199687847)

[*7.Export* 14](#_Toc199687848)

[*8.Modificare date* 14](#_Toc199687849)

[*9. Ștergere date* 15](#_Toc199687850)

[PLANIFICAREA PROIECTULUI 15](#_Toc199687851)

[Diagrama Gantt cu alocarea timpului de lucru pentru sarcini 15](#_Toc199687852)

[Repartizarea task-urilor între membrii echipei: 15](#_Toc199687853)

[RISCURI SI PROBLEME 16](#_Toc199687854)

[*1. Probleme organizaționale* 16](#_Toc199687855)

[*2. Probleme legate de date* 16](#_Toc199687856)

[*3. Probleme de securitate și confidențialitate* 16](#_Toc199687857)

[**IMPLEMENTAREA** 17](#_Toc199687858)

[1.GENERALITĂȚI 17](#_Toc199687859)

[1.1 *Denumire:* 17](#_Toc199687860)

[1.2 *Contextul proiectului:* 17](#_Toc199687861)

[1.3 *Cod:* 17](#_Toc199687862)

[1.4 *Scurtă descriere:* 17](#_Toc199687863)

[1.5 *Arhitectura sistemului:* 17](#_Toc199687864)

[1.6 *Funcții testate:* 18](#_Toc199687865)

[2.MIJLOACE DE VERIFICARE 18](#_Toc199687866)

[*Aplicații și unelte software* 18](#_Toc199687867)

[*Sisteme de baze de date și tehnologii* 18](#_Toc199687868)

[3.CONDIȚII DE FUNCȚIONARE 19](#_Toc199687869)

[*1.Colectare date medicale primare* 19](#_Toc199687870)

[*2.Gestionare și integrare date medicale* 19](#_Toc199687871)

[*3.Analiză clinică și alertare automate* 19](#_Toc199687872)

[*4.Dashboard pentru decizii rapide* 19](#_Toc199687873)

[*5.Modul blockchain* 19](#_Toc199687874)

[4.CONDIȚII DE VERIFICARE 20](#_Toc199687875)

[*Colectare date medicale primare* 20](#_Toc199687876)

[*Gestionare și integrare date medicale* 20](#_Toc199687877)

[*Analiză clinică și alertare automată* 20](#_Toc199687878)

[*Dashboard pentru decizii rapide* 20](#_Toc199687879)

[*Modul blockchain* 21](#_Toc199687880)

[5.CONFIGURARE ȘI INSTALARE 21](#_Toc199687881)

[*Dependențe:* 21](#_Toc199687882)

[*Rulare:* 21](#_Toc199687883)

[**TUTORIAL** 22](#_Toc199687884)

[1. LOGARE 22](#_Toc199687885)

[2. MONITORIZARE PACIENȚI 23](#_Toc199687886)

[3. ADĂUGARE PACIENT 25](#_Toc199687887)

[4. ADĂUGARE REZULTATE DE LABORATOR 25](#_Toc199687888)

[5. CALCULARE SCOR CLINIC 26](#_Toc199687889)

[6. MODIFICARE DATE 28](#_Toc199687890)

[7. ȘTERGERE DATE 29](#_Toc199687891)

[8. EXPORT DATE 32](#_Toc199687892)

[9. VIZUALIZARE DATE 33](#_Toc199687893)

# **PLANIFICAREA**

Această aplicație informatică a fost concepută pentru a sprijini chirurgia pediatrică de urgență prin colectarea, gestionarea și analiza datelor medicale primare ale pacienților. În cazurile de urgență, este esențial ca medicii să aibă acces rapid la informațiile relevante pentru a lua decizii corecte și rapide. Aplicația centralizează toate datele clinice într-o platformă ușor de folosit, permițând medicilor să acceseze datele pacienților, să înregistreze teste de laborator și să calculeze scoruri clinice și să primească alerte în timp real.

Pe lângă gestionarea datelor clinice, aplicația include și un modul de alertare bazat pe praguri clinice, care ajută la identificarea rapidă a riscurilor. În plus, toate datele sunt documentate și stocate într-un mod sigur, iar accesul și acțiunile medicilor sunt trasabile datorită integrării tehnologiei blockchain, asigurând transparență și siguranță în utilizare.

## SCOP

Scopul acestei aplicații este de a oferi un instrument eficient pentru gestionarea și analiza datelor medicale în timpul urgențelor chirurgicale pediatrice. Aplicația are rolul de a centraliza informațiile esențiale, de a automatiza procesele de alertare în caz de risc și de a sprijini luarea deciziilor clinice. Prin integrarea unui modul blockchain, aplicația asigură trasabilitatea și siguranța datelor, contribuind astfel la îmbunătățirea calității îngrijirii medicale. Totodată, aplicația ajută medicii să reacționeze rapid și eficient, reducând timpul necesar pentru diagnostic și intervenție, ceea ce este crucial în situațiile de urgență.

CONSTRANGERI DE PROIECTARE

- protecția datelor sensibile prin criptare și controlul accesului la date prin roluri bazate pe permisiuni;

- interfață cu un design intuitiv și introducere rapidă a datelor în condiții de urgență;

- performanța în timp real pentru suport decizional rapid.

## ARHITECTURA SISTEMULUI

Aplicația este construită modular, utilizând o arhitectură client-server, în care datele sunt colectate, procesate și vizualizate în timp real. Sistemul suportă atât preluarea automată a datelor medicale, cât și introducerea manuală, urmând un flux sigur și trasabil. Interoperabilitatea, trasabilitatea, analiza clinică automată și vizualizarea intuitivă sunt priorități

fundamentale.

Componentele principale ale arhitecturii includ:

* Frontend: Interfața utilizatorului, creată cu Streamlit, permite medicilor să introducă datele pacientului și să vizualizeze rezultatele analizei în timp real.
* Backend: Folosește FastAPI pentru a gestiona cererile HTTP de la frontend și pentru a procesa datele.
* Baze de date: Utilizarea PostgreSQL pentru stocarea datelor structurate
* Analiză și vizualizare: Pandas și Plotly sunt folosite pentru analiza datelor și generarea de grafice.
* Blockchain: Folosirea web3.py și Ganache pentru logarea evenimentelor într-un sistem blockchain pentru trasabilitate și securitate.

## PROIECTARE DETALIATA – HARDWARE SI SOFTWARE

Corelație dintre Componente și Tehnologii:

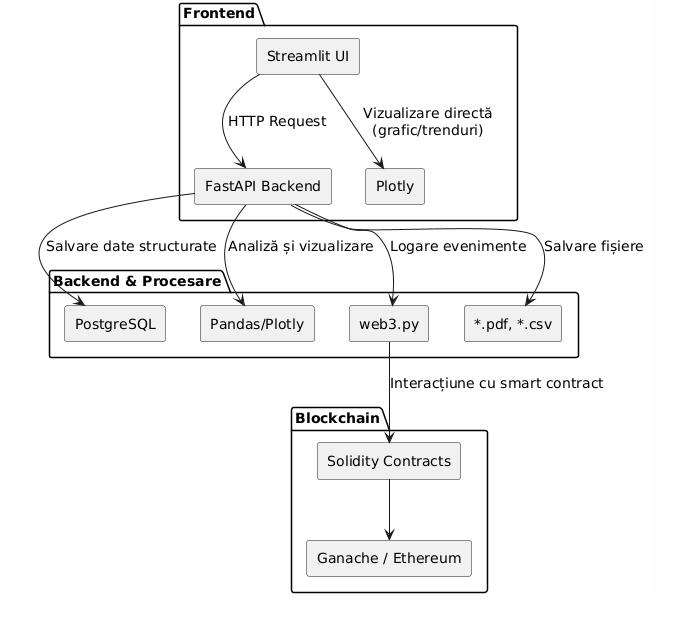
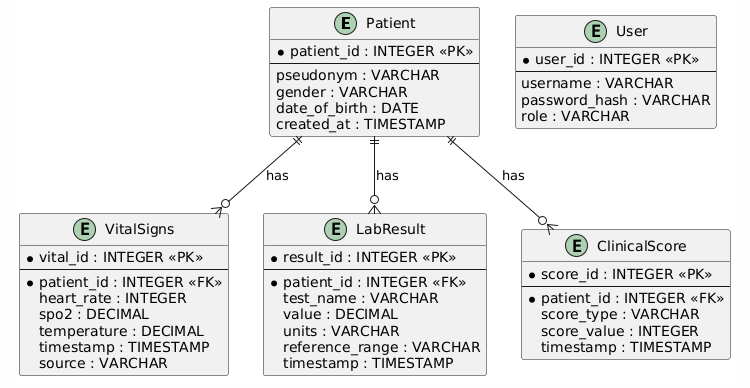


Diagrama de mai sus ilustrează cum colaborează tehnologiile în cadrul aplicației. Interfața cu utilizatorul este realizată în Streamlit și interacționează cu backend-ul (FastAPI). Backend-ul salvează datele pacientului în PostgreSQL (pentru date structurate). Datele sunt trimise apoi către Pandas și Plotly pentru a fi analizate și vizualizate. În același timp, toate evenimentele importante sunt înregistrate într-un sistem blockchain pentru a asigura trasabilitatea și securitatea acestora, utilizând web3.py și Ganache.

## PROIECTAREA BAZEI DE DATE

Sistemul utilizează o arhitectură relațională, bazată pe PostgreSQL, pentru gestionarea datelor medicale structurate din chirurgia pediatrică de urgență. Structura bazei de date este centrată pe entitatea principală Patient, în jurul căreia gravitează toate celelalte entități relevante, inclusiv semne vitale, rezultate de laborator, scoruri clinice și loguri de acces.



Tabele principale:

Patient - Conține date de identificare ale pacienților

* + patient\_id - identificator numeric unic
  + pseudonym, gender, date\_of\_birth - informații esențiale
  + created\_at - dată înregistrare pacient

VitalSigns - Stochează date despre semnele vitale, asociate unui patient\_id:

* + heart\_rate - pulsul,
  + spo2 – concentrația de oxigen din sânge,
  + temperature - temperatura,
  + timestamp - ora înregistrării,
  + source - sursa din care provine (manual, automat, simulat)

LabResult - Rezultate de laborator standardizate:

* + test\_name - tipul de test efectuat,
  + value - valoarea in urma testului,
  + units - unitățile,
  + reference\_range - intervalul de referință,
  + timestamp - data și ora înregistrării

ClinicalScore - Scoruri clinice aplicate pacienților (ex. PEWS, NEWS, GCS Pediatric, PRISM, pSOFA):

* + score\_type - tipul scorului (string)
  + score\_value - valoare numerică
  + timestamp - momentul evaluării

User - Utilizatori ai platformei (medici, asistenți, cercetători): -

* + username - numele utilizatorului,
  + hashed\_password - parola cryptată,
  + role - rolul utilizatorului

Datele pot fi extrase în format CSV sau JSON, pentru a fi utilizate în aplicații externe (ex. R, SPSS, Power BI) sau analizate în scopuri științifice, clinice sau de cercetare.

Structura relațională:

Toate datele vitale, scorurile clinice și analizele de laborator sunt legate de un pacient unic (patient\_id) prin relații de tip Foreign Key. Fiecare utilizator are acțiunile sale logate într-un contract blockchain, asigurând trasabilitate și controlul auditului.

## REFERINTE DE PROIECT

• European Paediatric Surgeons’ Association (EUPSA) – Organizație care promovează standarde înalte de îngrijire chirurgicală pediatrică în Europa. https://emedic.ro/curriculum-de-pregatire-in-specialitatea-chirurgie-pediatrica

• Protocoale terapeutice – Pediatrica.ro – Inițiativă pentru elaborarea și acceptarea normelor terapeutice în chirurgia și ortopedia pediatrică din România. https://pediatrica.ro/protocoale-terapeutice/

• Societatea Română de Chirurgie Pediatrică (SRCP) – Organizație profesională care promovează standarde clinice și științifice în chirurgia pediatrică din România. https://srcp.ro/

• Documentatia oficiala FastAPI. https://fastapi.tiangolo.com

• Documentatia oficiala Streamlit. <https://docs.streamlit.io>

## SECURITATEA SI INTEGRITATEA PROIECTULUI

1. Autentificare și autorizare (FastAPI + JWT)

Sistemul asigură autentificarea utilizatorilor și controlul accesului prin:

* + Generare token JWT securizat (algoritm HS256);
  + Parole criptate cu bcrypt;
  + Decoratori pentru restricționarea rutelor (@role\_required([...]));
  + Posibilitatea de integrare cu 2FA;

Roluri și permisiuni:

* + Medic – acces complet: introducere, analiză, export;
  + Asistent – vizualizare și completare date;
  + Cercetător – acces doar la date anonime și export.

2. Stocare securizată și protecție date

Baze de date:

* + PostgreSQL – date structurate;
  + Identificator unic pseudonimizat pentru fiecare pacient;
  + Criptare AES-256 aplicată datelor sensibile;

3. Trasabilitate și logare evenimente în blockchain

Orice acțiune medicală importantă (de ex. adăugare de date) este înregistrată criptografic și imutabil într-un smart contract Ethereum.

4. Dashboard medical cu alertă și log automat în blockchain

Interfața UI dashboard monitorizează în timp real datele vitale și semnalează alertele critice

5. Export securizat pentru analiză științifică (JASP, R)

Funcționalități integrate:

* + Export CSV pentru JASP;
  + Export JSON pentru R;

# **ANALIZA**

## CERINTE DE PROIECT

### *1. Cerințe funcționale:*

* Sistemul trebuie să permită colectarea datelor medicale din surse variate (API HL7/FHIR, introducere manuală).
* Datele colectate trebuie validate și stocate securizat în format standardizat.
* Sistemul trebuie să analizeze datele în timp real și să emită alerte medicale în cazuri critice.
* Interfața grafică (dashboard) trebuie să permită vizualizarea datelor în timp real
* Toate acțiunile asupra datelor trebuie înregistrate într-un registru blockchain pentru auditabilitate.

### *2. Cerințe nefuncționale:*

* Sistemul trebuie să respecte standardele de confidențialitate (GDPR).
* Datele trebuie criptate cu AES-256.
* Autentificarea utilizatorilor se va face cu autentificare în doi pași (2FA).
* Aplicația trebuie să asigure redundanță și disponibilitate ridicată.
* Interfața trebuie să fie intuitivă, cu timp de răspuns sub 2 secunde pentru afișarea datelor.

## CAZURI DE UTILIZARE

### *1.Login*

|  |  |
| --- | --- |
| Nume | Login |
| ID | 1 |
| Descriere | Autentificarea utilizatorilor în aplicație prin nume de utilizator și parolă. |
| Actori | Utilizator (medic, asistent, cercetător)  Sistem (FastAPI, baza de date users) |
| Preconditii | Utilizatorul trebuie să existe în baza de date (users)  Parola trebuie să fie hash-ată corect  Backend-ul FastAPI trebuie să fie funcțional |
| Postconditii | Se generează un token JWT valid  Utilizatorul primește acces la interfață conform rolului |
| Flux principal | 1. Utilizatorul introduce credențialele 2. Sistemul verifică în baza de date 3. Dacă sunt valide, returnează un token JWT |
| Exceptii | Username sau parolă incorectă  Server indisponibil |

### *2.Adaugare pacient*

|  |  |
| --- | --- |
| Nume | Adaugare Pacient |
| ID | 2 |
| Descriere | Permite adăugarea unui nou pacient în sistem, cu date personale anonimizate |
| Actori | Medic / Asistent  Sistem (FastAPI, PostgreSQL) |
| Preconditii | Utilizatorul este autentificat  Are rol de medic sau asistent |
| Postconditii | Se creează o înregistrare în tabela patients |
| Flux principal | 1. Utilizatorul completează formularul (pseudonim, sex, data nașterii) 2. Sistemul creeaza pacientul in patients |
| Exceptii | Lipsă câmp obligatoriu |

### *3.Adaugare rezultate laborator*

|  |  |
| --- | --- |
| Nume | Adaugare rezultate de laborator |
| ID | 3 |
| Descriere | Înregistrarea valorilor de laborator pentru un pacient existent (ex: leucocite, CRP etc.) |
| Actori | Medic / Asistent  Sistem (FastAPI, PostgreSQL) |
| Preconditii | Utilizatorul este autentificat  Are rol de medic sau asistent |
| Postconditii | Se creează o înregistrare în tabela LabResults |
| Flux principal | 1. Se selectează pacientul 2. Se introduc datele testului (nume, valoare, unitate, interval) 3. Se salvează cu timestamp curent |
| Exceptii | Lipsă câmp obligatoriu  Pacient inexistent |

### *4.Calculare scor clinic*

|  |  |
| --- | --- |
| Nume | Calculare scor clinic |
| ID | 4 |
| Descriere | Permite calculul automatizat al unui scor clinic (PEWS, NEWS, pSOFA etc.) |
| Actori | Medic  Sistem (FastAPI, PostgreSQL) |
| Preconditii | Utilizatorul este autentificat  Are rol de medic. |
| Postconditii | Scorul calculat poate fi afișat și salvat în ClinicalScore |
| Flux principal | 1. Utilizatorul alege pacientul și scorul 2. Introduce parametrii 3. Sistemul calculează scorul 4. Se afișează interpretarea |
| Exceptii | Lipsă câmp obligatoriu  Pacient inexistent |

### *5.Vizualizare date*

|  |  |
| --- | --- |
| Nume | Vziualizare date |
| ID | 5 |
| Descriere | Permite consultarea istoricului medical (semne vitale, scoruri, analize) pentru un pacient |
| Actori | Utilizator  Sistem (FastAPI, PostgreSQL) |
| Preconditii | Utilizator logat  Are drepturi de acces (rol: medic, asistent, cercetător) |
| Postconditii | Datele sunt afișate în interfața Streamlit |
| Flux principal | 1. Se selectează pacientul 2. Sistemul preia datele din vital\_signs, clinical\_scores, lab\_results 3. Se afișează tabelar sau grafic |
| Exceptii | Lipsă date  Probleme la conectarea bazei de date |

### *6.Monitorizare pacienti*

|  |  |
| --- | --- |
| Nume | Monitorizare |
| ID | 6 |
| Descriere | Vizualizare grafică a evoluției stării pacientului în timp (ex: SpO2, puls temperatură), alertare in cazul intrecerii pragurilor |
| Actori | Utilizator  Sistem |
| Preconditii | Utilizator logat  Are drepturi de acces  Există date multiple înregistrate pentru un pacient |
| Postconditii | Se afișează grafice interactive cu Plotly  Se afiseaza o alerta in cazul in care parametrii depasesc anumite praguri |
| Flux principal | 1. Se selectează pacientul/pacientii 2. Se alege parametrul (ex: HR, scor PEWS) 3. Se generează graficul |
| Exceptii | Lipsă date  Probleme la conectarea bazei de date |

### *7.Export*

|  |  |
| --- | --- |
| Nume | Export |
| ID | 7 |
| Descriere | Permite salvarea locală a datelor pacientului pentru raportare, cercetare sau backup |
| Actori | Utilizator  Sistem |
| Preconditii | Utilizatorul este logat și are permisiuni |
| Postconditii | Se generează fișiere .csv sau .json |
| Flux principal | 1. Utilizatorul alege pacientul 2. Utilizatorul selecteaza datele pe care doreste sa le exporte (datele pacientului, semne vitale, analize de laborator, scoruri clinice) 3. Sistemul preia datele și le scrie într-un fișier 4. Utilizatorul descarcă fișierul |
| Exceptii | Eroare de scriere fișier  Pacient fără date |

### *8.Modificare date*

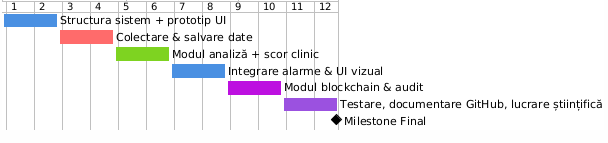
|  |  |
| --- | --- |
| Nume | Modificare date |
| ID | 8 |
| Descriere | Permite modificarea datelor personale sau a rezultatelor de laborator salvate in baza de date |
| Actori | Medic |
| Preconditii | Utilizatorul este logat și are permisiunile necesare  Pacientul ales este salvat si are date salvate în baza de date |
| Postconditii | Se modifică datele alese si se salvează modificarile in baza de date |
| Flux principal | 1. Utilizatorul alege pacientul 2. Utilizatorul selecteaza setul de date pe care dorește să le modifice 3. Sistemul preia modificările si le salvează |
| Exceptii | Eroare la salvare  Pacient fără date |

### *9. Ștergere date*

|  |  |
| --- | --- |
| Nume | Ștergere date |
| ID | 9 |
| Descriere | Permite ștergerea integrală a datelor pacientului, rezultatelor de laborator și scorurilor clinice salvate |
| Actori | Medic |
| Preconditii | Utilizatorul este logat și are permisiunile necesare |
| Postconditii | Se șterg datele alese din baza de date |
| Flux principal | 1. Utilizatorul alege pacientul 2. Utilizatorul selecteaza setul de date pe care doreste sa le șteargă 3. Sistemul șterge datele respective din baza de date |
| Exceptii | Date salvate greșit |

## PLANIFICAREA PROIECTULUI

### Diagrama Gantt cu alocarea timpului de lucru pentru sarcini

****

### Repartizarea task-urilor între membrii echipei:

* Lucaciu Mihai-Remus: team-leader, coordonează echipa; backend și baza de date, se asigură de calitatea produsului și integrarea modulelor dezvoltate de ceilalți membrii
* Iancu Mara-Ștefania: frontend; modulul dashboard cu alerte, modulul de modificare și ștergere a datelor
* Manea Roberta-Maria: frontend; modulul de adăugare date, rezultate de laborator, calculare scor clinic; integrare blockchain
* Leuca Romina-Anamaria: modulul de vizualizare și export de date; întocmirea documentației

## RISCURI SI PROBLEME

### *1. Probleme organizaționale*

-Lipsa respectării planului de lucru: echipa poate întârzia livrarea anumitor module dacă nu se respectă planificarea pe săptămâni.

-Distribuire inegală a sarcinilor: unii membri pot ajunge să aibă o încărcare disproporționată față de alții, ceea ce creează tensiuni și scade eficiența.

-Neînțelegeri tehnice între membri: dacă nu toți sunt familiarizați cu tehnologiile folosite (ex: blockchain), apar blocaje.

### *2. Probleme legate de date*

-Date incomplete sau incorecte: formularele pot fi completate greșit, ceea ce duce la erori în analiză.

-Date prea complexe: volume mari de date sau fișiere necorespunzătoare (ex: pdf-uri scanate) pot fi greu de procesat automat.

-Pierderea datelor: în lipsa unui backup sau criptare corespunzătoare, datele pot fi compromise.

### *3. Probleme de securitate și confidențialitate*

-Riscuri GDPR: colectarea și stocarea datelor medicale implică obligații legale stricte.

-Acces neautorizat: dacă autentificarea 2FA nu e implementată corect, datele pot fi expuse.

# **IMPLEMENTAREA**

## 1.GENERALITĂȚI

* 1. *Denumire:*  
     Sistem informatic pentru colectarea, gestionarea și analiza datelor medicale primare în chirurgia pediatrică de urgență.
  2. *Contextul proiectului:*  
     Aplicația este dezvoltată ca instrument suport în luarea deciziilor clinice rapide, prin centralizarea și analiza automată a datelor medicale.

### *Cod:*

Codul sursă este implementat în Python și structurat în două componente principale:

* app/ – backend FastAPI
* dashboard/ – interfață Streamlit
  1. *Scurtă descriere:*Aplicația informatică pentru chirurgia pediatrică de urgență este un sistem inteligent destinat colectării, gestionării și analizării datelor medicale ale pacienților. Concepută pentru a sprijini deciziile rapide și corecte ale medicilor în situații critice, aplicația oferă acces centralizat la istoricul medical, permite înregistrarea datelor, a rezultatelor de laborator, calcularea scorurilor clinice și generează alerte automate bazate pe praguri clinice. Prin integrarea tehnologiei blockchain, asigură trasabilitatea acțiunilor și securitatea datelor, contribuind astfel la eficientizarea intervențiilor și creșterea calității actului medical. Sistemul integrează module pentru introducere date (Streamlit), procesare backend (FastAPI), stocare (PostgreSQL), analiză (Pandas, Plotly) și trasabilitate (Blockchain - Ganache & Web3.py).

### *Arhitectura sistemului:*

* Frontend: Streamlit
* Backend: FastAPI
* DB: PostgreSQL (date structurate)
* Analiză: Pandas, Plotly
* Trasabilitate: Blockchain

### *Funcții testate:*

* Autentificare și autorizare
* Introducere date pacient
* Modificarea datelor
* Ștergerea datelor
* Vizualizare grafică evoluție
* Emitere alertă
* Export JSON/CSV
* Logare evenimente în blockchain

## 2.MIJLOACE DE VERIFICARE

### *Aplicații și unelte software*

* IDE: Visual Studio Code / PyCharm
* Browsere: Chrome, Firefox, Edge (pentru testare interfață)
* Instrumente de testare API: Postman, Swagger UI
* Dashboard: Streamlit, Dash, Plotly

### *Sisteme de baze de date și tehnologii*

* PostgreSQL – pentru date structurate
* FastAPI – backend
* Python (Flask/FastAPI) – analiză date și scoruri
* Ethereum Testnet / Ganache – logare acțiuni blockchain
* HL7/FHIR – pentru interoperabilitate date medicale

## 3.CONDIȚII DE FUNCȚIONARE

Mai jos sunt descrise toate funcționalitățile care trebuie testate pentru validarea sistemului:

### *1.Colectare date medicale primare*

* + Formularul de colectare poate primi date manual.
  + Datele sunt validate automat și criptate.
  + Utilizatorul primește feedback de succes sau eroare.

### *2.Gestionare și integrare date medicale*

* + Datele sunt salvate corect în PostgreSQL.
  + Sistemul recunoaște pacienții după ID unic și evită duplicatele.
  + Integrarea între module este sincronizată, iar fiecare acțiune este logată.

### *3.Analiză clinică și alertare automate*

* + Sistemul analizează în timp real parametrii
  + Emiterea alertelor se face când se depășesc pragurile critice.

### *4.Dashboard pentru decizii rapide*

* + Se afișează în timp real datele pacientului și valorile critice.
  + Interfața este accesibilă doar personalului autorizat.

### *5.Modul blockchain*

* + Orice acces sau modificare a datelor este logată în blockchain.
  + Se generează o tranzacție cu timestamp și hash.

## 4.CONDIȚII DE VERIFICARE

Această secțiune explică cum se validează concret fiecare funcționalitate descrisă în secțiunea precedentă:

### *Colectare date medicale primare*

* + Se introduc manual date de test pentru un pacient fictiv.
  + Se verifică:
    - criptarea datelor în baza de date,
    - logurile de validare.
  + Se confirmă feedback-ul (success/error) primit de utilizator.

### *Gestionare și integrare date medicale*

* + Se trimit date de test către sistem și se verifică stocarea lor.
  + Se urmărește logul de audit pentru confirmarea acțiunilor.
  + Se verifică că nu sunt acceptate ID-uri duplicate.
  + Se simulează o eroare de conectare și se observă notificarea sistemului.

### *Analiză clinică și alertare automată*

* + Se trimit parametri vitale care depășesc pragurile critice.
  + Se verifică emiterea unei alerte vizuale/audio.

### *Dashboard pentru decizii rapide*

* + Se accesează dashboard-ul dintr-un cont medical și se verifică afișarea corectă a graficelor în timp real.
  + Se testează accesul unui utilizator fără drepturi (eroare de acces).

### *Modul blockchain*

* + Se modifică datele unui pacient test și se urmărește înregistrarea tranzacției în Ganache.
  + Se validează prezența timestamp-ului și hash-ului în blockchain.
  + Se testează un rol neautorizat pentru a confirma blocarea accesului.

## 5.CONFIGURARE ȘI INSTALARE

### *Dependențe:*

Pentru funcționarea corectă a aplicației următoarele librării python trebuie instalate: fastapi, pydantic, pandas, passlib, sqlalchemy, jose, web3, solcx, streamlit, plotly, uvicorn

Rulează următoarea comandă în terminal:

*pip install fastapi, pydantic, pandas, passlib, sqlalchemy, jose, web3, solcx, streamlit, plotly, uvicorn*

### *Rulare:*

Pentru a folosi aplicația este nevoie să fie pornit un server PostgreSQL (pentru baza de date) si un server Ganache (pentru contractul blockchain).

De asemenea, este nevoie să fie pornit un server de backend.

1. Navighează în terminal folosind comanda *cd* către folderul unde este instalată aplicația
2. Rulează următoarea comandă

*uvicorn app.main:app –reload*

Serverul de backend este pornit.

1. Apoi, deschide un alt terminal, navighează din nou în director și rulează următoarea comandă:

*streamlit run dashboard/home.py*

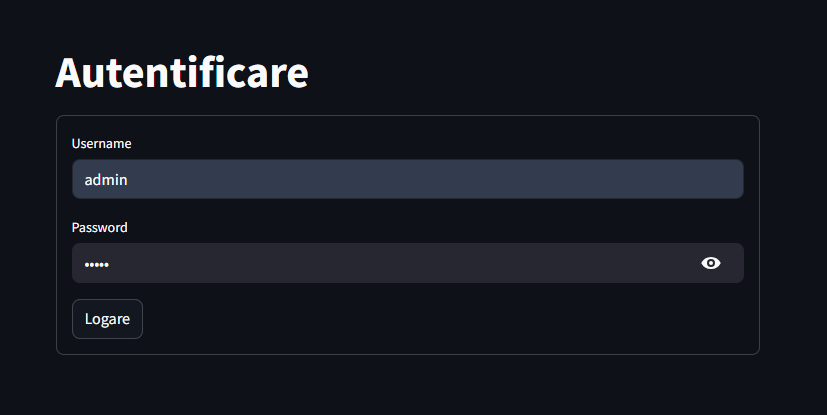
Va trebui să se deschidă o fereastră nouă in browser-ul tău. Aplicația rulează.

# **TUTORIAL**

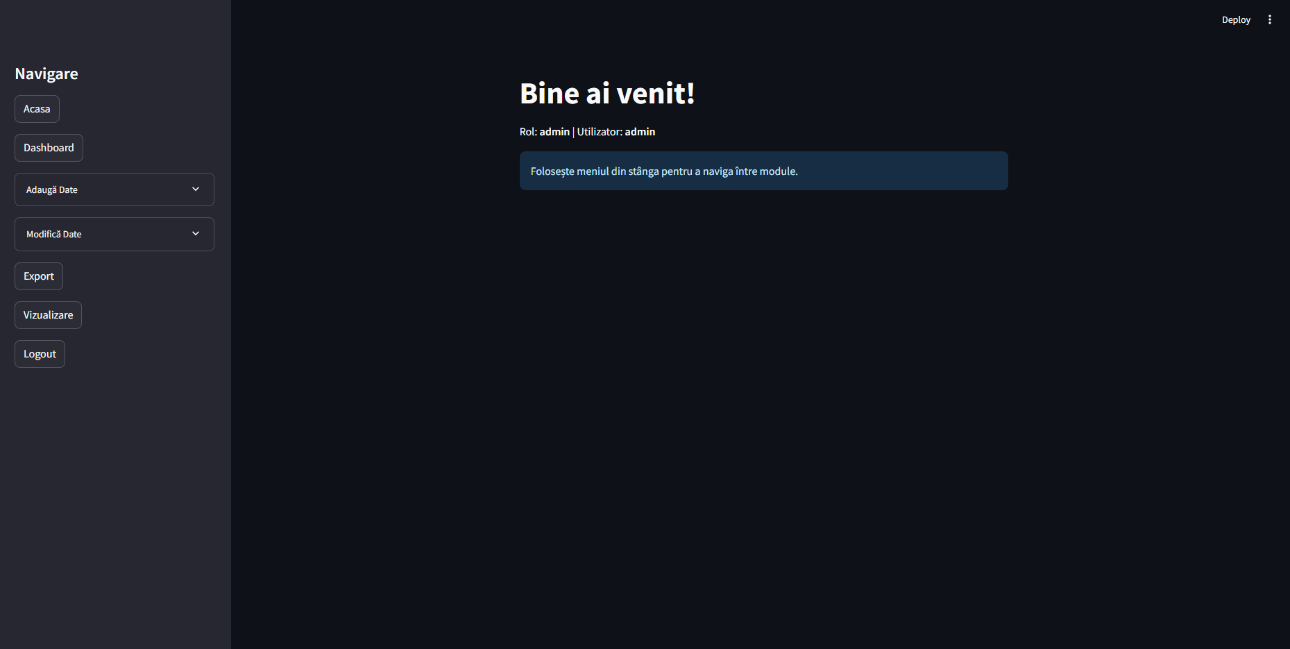
## LOGARE

Pentru a folosi aplicația, mai intâi este nevoie să o rulezi. Vezi Configurare și Instalare ca să afli cum.

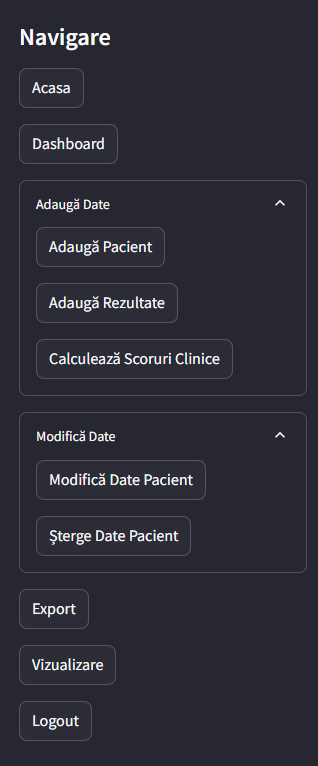
După ce aplicația a fost pornită, se va deschide in browser următoarea pagină



Introduce numele de utilizator in căsuța Username și parola în căsuța Password, iar apoi apasă pe butonul Logare. Dacă ai introdus credențialele corecte, te vei conecta la cont și vei fi întâmpinat de următoarea pagină



În partea din stânga este bara de navigare, de unde poți accesa toate modulele.



## MONITORIZARE PACIENȚI

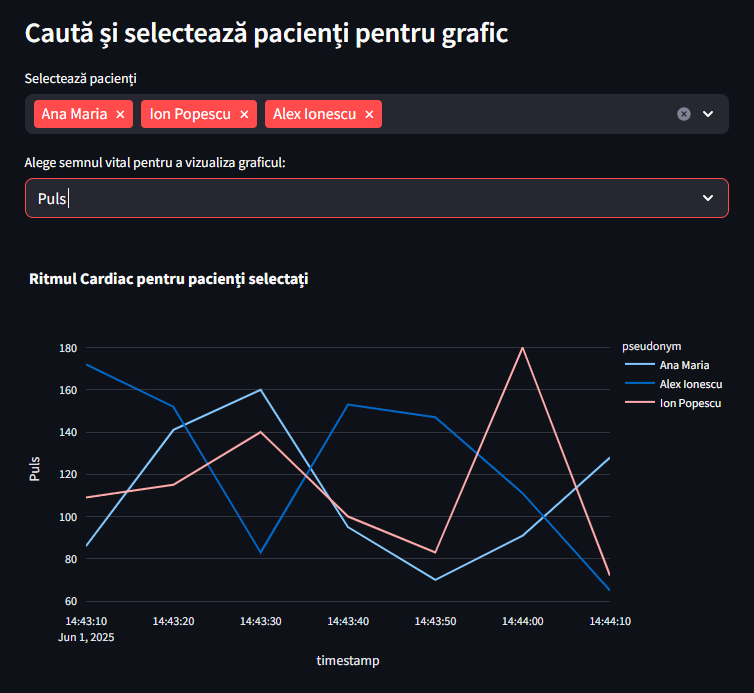
Pentru a monitoriza pacienții, apasă pe butonul Dashboard din bara de navigare. Se va deschide următoarea fereastră.

În partea de sus vei vedea alertele critice. Dacă valoarea pentru un semn vital depășește un anumit prag, atunci se va emite o alertă pentru pacientul și semnul vital respectiv.

Mai jos, poți vizualiza evoluția semnelor vitale ale pacienților pe un grafic. Pentru a vizualiza graficul, trebuie selectat minim un pacient. După selectarea pacientului, va fi afișat graficul. Poți selecta care semn vital vrei sa monitorizezi pe grafic.

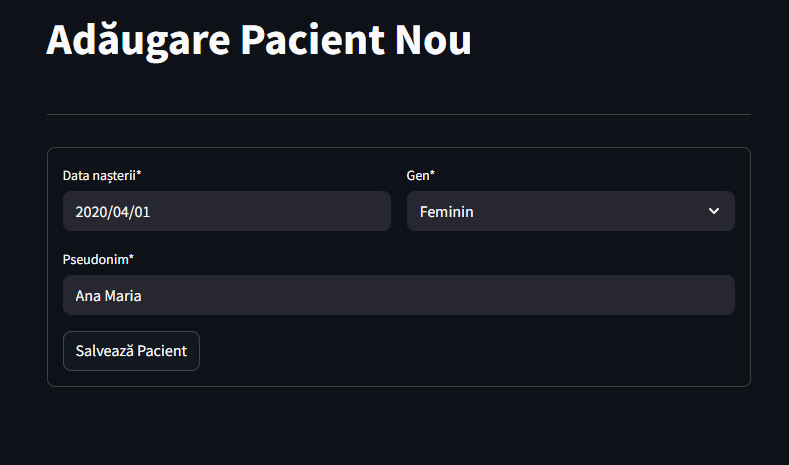
Mai jos, poți vizualiza grafice separate pentru fiecare pacient selectat.





## ADĂUGARE PACIENT

Pentru a adăuga un pacient nou în baza de date, apasă pe butonul Adaugă pacient din bara de navigare. Se va deschide următoarea pagină



Completează datele din formular si apasă pe butonul Salvează Pacient pentru a salva pacientul în baza de date.

Pentru a putea adăuga pacienți noi în baza de date, ai nevoie de permisiunile necesare (medic, asistent).

## ADĂUGARE REZULTATE DE LABORATOR

Pentru a adăuga rezultate de laborator pentru un pacient, apasă pe butonul Adaugă Rezultate din bara de navigare. Se va deschide urmatoarea fereastră.

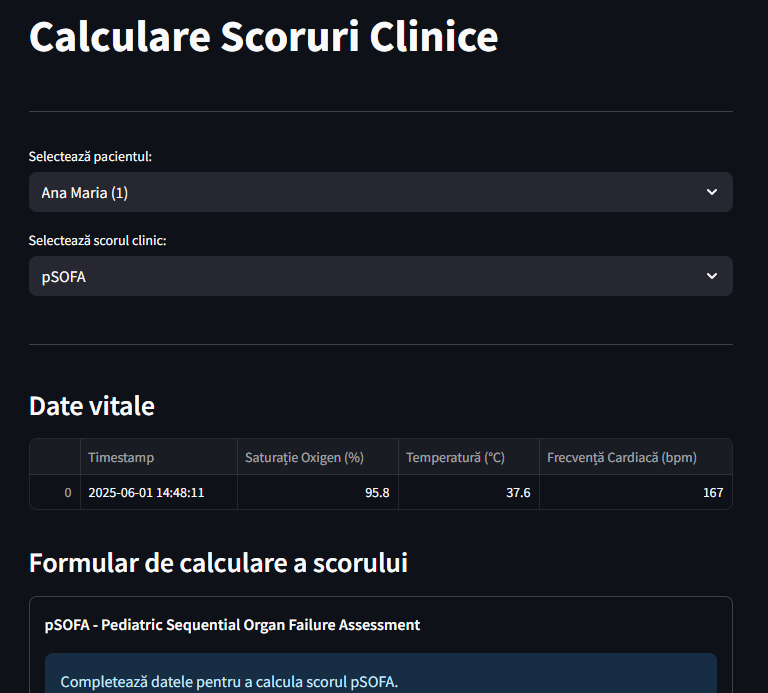
Selectează pacientul pentru care vrei sa adaugi rezultate de laborator, completează formularul și apasă pe butonul Salvează Rezultat pentru a salva analizele în baza de date.

Pentru a adăuga rezultate de laborator, ai nevoie de permisiunile necesare (medic, asistent)

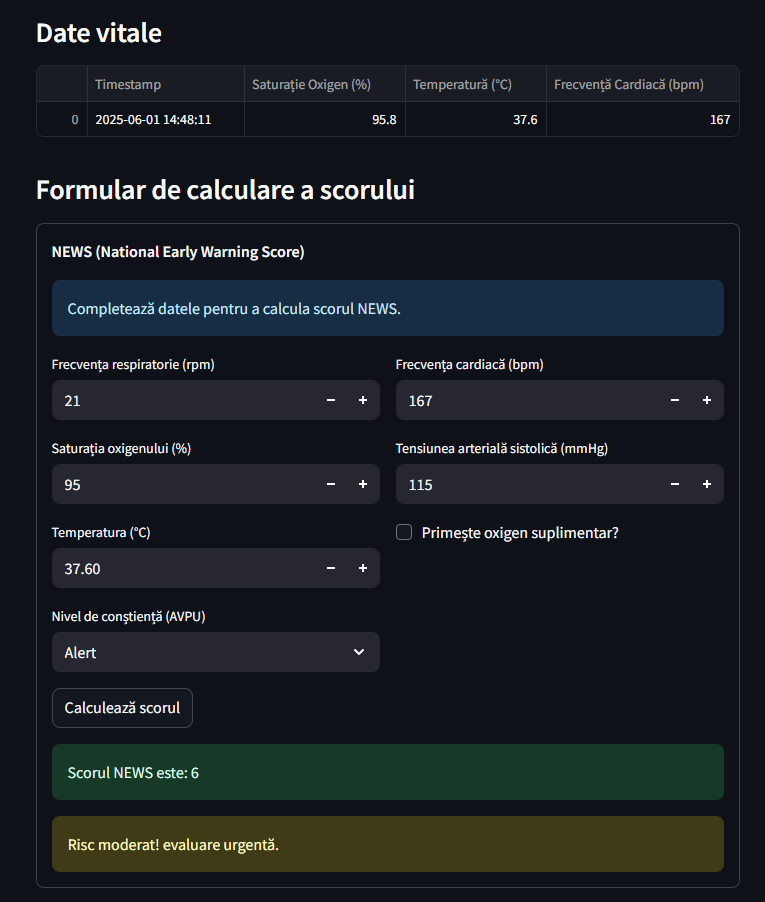


## CALCULARE SCOR CLINIC

Pentru a calcula un scor clinic, apasă pe butonul Calculează Scoruri Clinice din bara de navigare. Următoarea pagină se va deschide



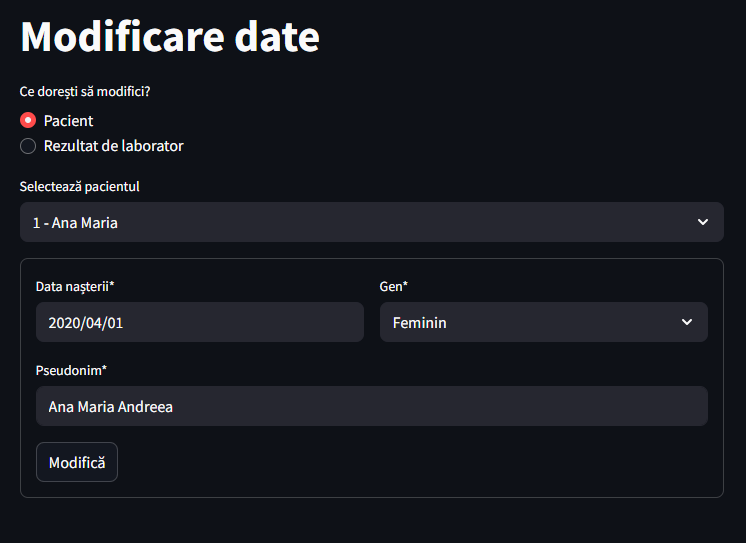
Selectează pacientul pentru care vrei să calculezi un scor clinic. Apoi selectează tipul de scor pe care vrei să îl calculezi. Completează formularul și apasă pe butonul Calculează Scor. Scorul va fi calculat si afișat, împreună cu o interpretare a rezultatului. Scorul va fi salvat în baza de date.



Pentru a putea calcula scorul clinic, ai nevoie de permisiunile necesare (medic, asistent)

## MODIFICARE DATE

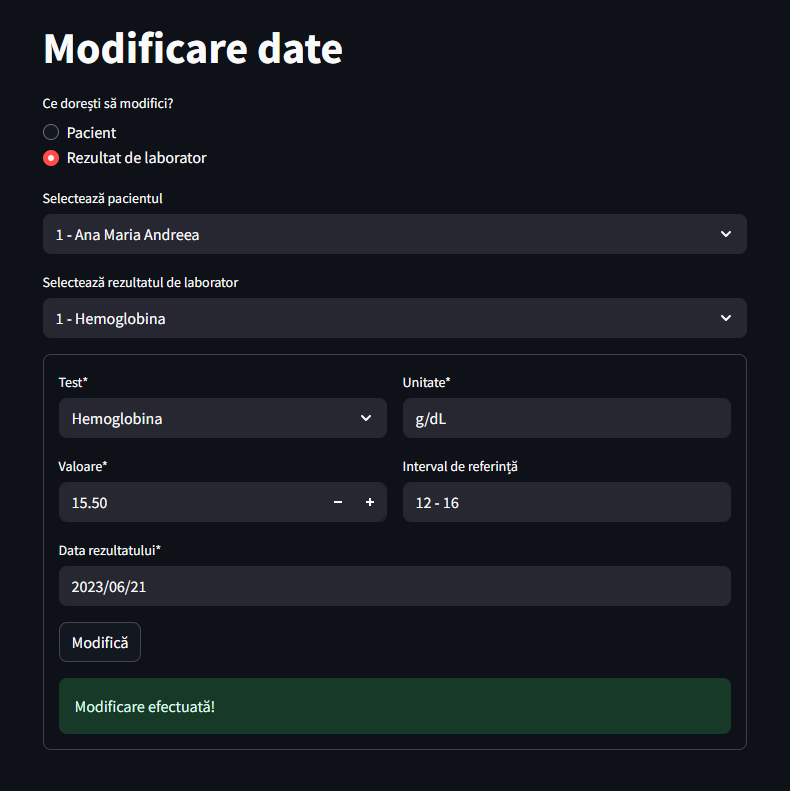
În cazul în care s-a produs o greșeală la introducerea datelor, este posibilă modificarea lor. Pentru a modifica datele salvate, apasă pe butonul Modifică Date Pacient din bara de navigare. Se va deschide următoarea fereastră.



Aici poți alege dacă dorești să modifici datele pacientului sau rezultatele de laborator, apăsând pe butoanele respective.

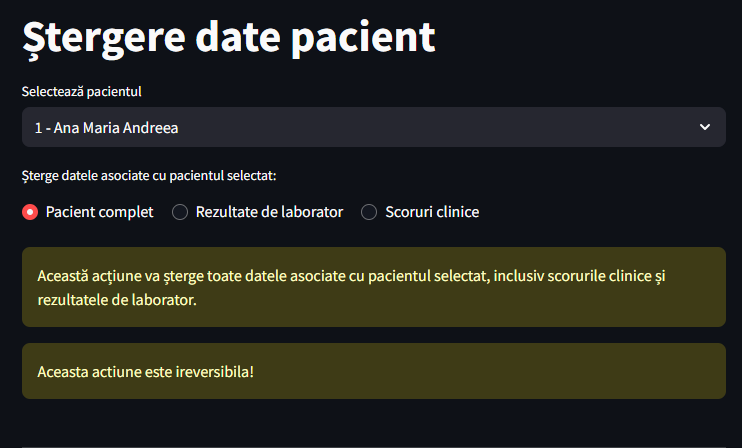
Selectează pacientul pentru care dorești sa modifici date. În cazul în care dorești sa modifici rezultate de laborator, selectează rezultatul pe care dorești să îl modifici. Completează formularul cu datele noi și apasă pe butonul Modifică. Datele vor fi modificate în baza de date.

Pentru a putea modifica date, ai nevoie de permisiunile necesare (medic)



## ȘTERGERE DATE

Pentru a șterge date din baza de date, apasă pe butonul Șterge Date Pacient din bara de navigare. Se va deschide următoarea pagină



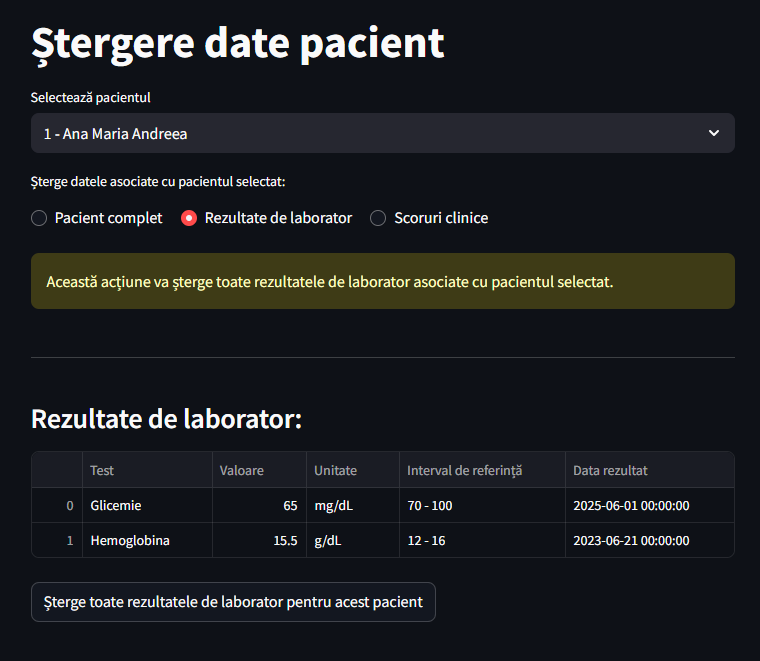
Selectează pacientul pentru care dorești să ștergi date. Apoi apasă pe butonul corespunzător datelor pe care dorești să le ștergi.

ATENȚIE! ȘTERGEREA DATELOR ESTE IREVERSIBILĂ!

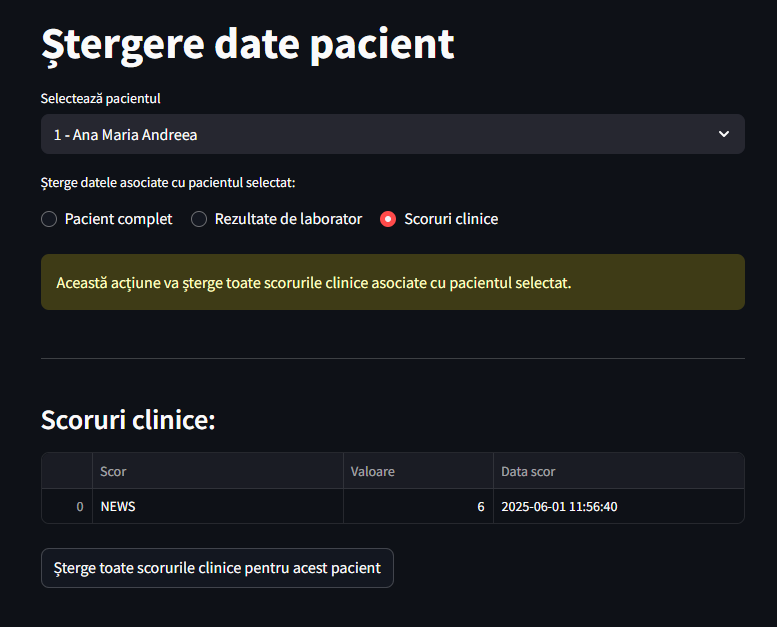
Dacă dorești să ștergi pacientul selectat si toate datele associate din baza de date, inclusiv rezultatele de laborator, scorurile clinice, si semnele vitale salvate, apasă pe butonul Pacient complet. Vor fi afișate toate datele associate cu pacientul selectat. Pentru a le șterge, apasă pe butonul Șterge toate datele asociate cu acest pacient.



Pentru a șterge doar rezultatele de laborator asociate unui pacient, apasă pe butonul Rezultate de laborator. Vor fi afișate toate rezultatele salvate în baza de date. Apasă butonul Șterge toate rezultatele de laborator pentru acest pacient pentru a le șterge.



Pentru a șterge doar scorurile clinic salvate, apasă pe butonul Scoruri clinice. Vor fi afișate toate scorurile salvate. Apasă pe butonul Șterge toate scorurile clinice pentru acest pacient pentru a le șterge.



Pentru a șterge date, ai nevoie de permisiunile necesare (medic)

## EXPORT DATE

Pentru a exporta date, apasă pe butonul Export din bara de navigare. Se va deschide următoarea fereastră



Selectează pacientul a carui date dorești sa exportezi. Dă click pe fiecare căsuță pentru a exporta datele respective.

Date pacient – datele pacientului (nume, data nasterii, gen)

Date vitale – ultimele date vitale înregistrate (temperatură, puls, SpO2)

Rezultate de laborator – toate rezultatele de laborator associate pacientului

Scoruri clinic – toate scorurile clinic calculate pentru pacient

Jurnal de acces – log-urile de acces asupra pacientului (inregistrare, modificare, etc)

După ce ai selectat ce date dorești să exportezi, selectează formatul în care dorești să exportezi datele dând click pe butonul respective.

Apasă pe butonul Descarcă CSV / Descarcă JSON pentru e a descărca un fișier .csv / .json care conține datele selectate asociate pacientului ales.

## VIZUALIZARE DATE

Pentru a vizualiza toate datele unui pacient, apasă pe butonul Vizualizare din bara de navigare. Se va deschide următoarea pagină

Aici vei vedea o listă cu toți pacienții salvati în baza de date. În bara de cautare, poți căuta un pacient anume.

Sub lista pacienților, poți selecta ID-ul unui pacient pentru a vizualiza toate datele salvate asociate pacientului respectiv.

