

Aplicatie E-Learning - UpLearn

Adrian Vlose

1 Arhitectura conceptuala

Diagrama conceptuala prezinta fluxul logic dintre interfata utilizatorului (Student/Profesor) si microserviciile de backend. Sistemul este compus din doua microservicii principale:

- **User Management Microservice** - gestioneaza informatiile legate de utilizatori (studenti si profesori) si interactioneaza cu baza de date *User Database*.
- **Course Management Microservice** - gestioneaza informatiile legate de cursuri si interactioneaza cu baza de date *Course Database*.

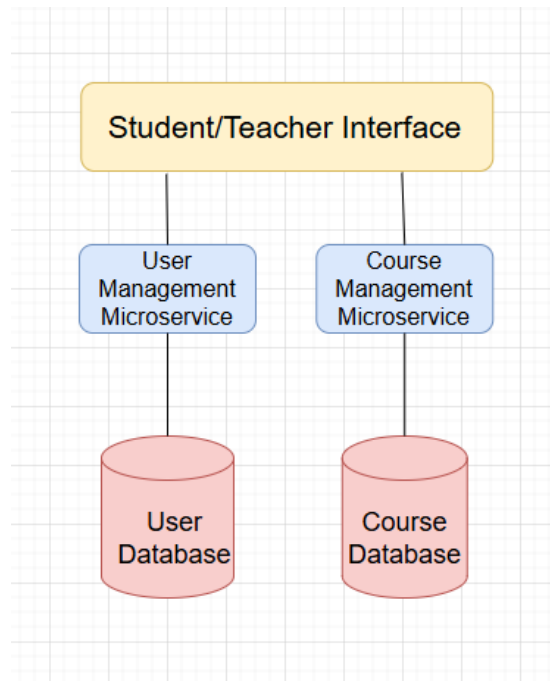


Figure 1: Diagrama Conceptuala a sistemului

2 Arhitectura tehnica

Arhitectura tehnica este implementata folosind un stack MERN partial (fara MongoDB), astfel:

- **Frontend:** React - responsabil pentru interfata utilizatorului.
- **Backend:** Node.js impreuna cu Express - gestioneaza request-urile si logica de server.
- **Baza de date:** MySQL - pastreaza datele aplicatiei.

Frontend-ul comunica prin API REST cu backend-ul Node.js, iar acesta interactioneaza cu baza de date MySQL.

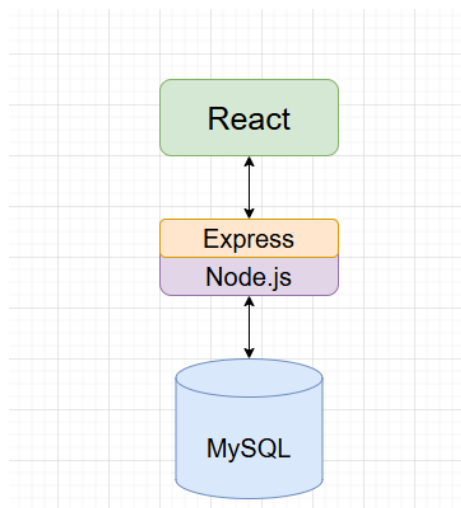


Figure 2: Diagrama tehnica a sistemului

3 Modelul de date

Modelul de date al aplicatiei este impartit in doua baze de date separate: **User Database** si **Course Database**.

- **User Database** contine tabela *Users*, responsabila de gestionarea datelor utilizatorilor, precum username, parola, email, telefon si rolul acestora.
- **Course Database** include tabelele *Courses*, *Discount*, *Enrollments* si *UserRoles*.

3.1 Course Database

Course Database contine urmatoarele tabele esentiale pentru gestionarea cursurilor si a inscrierilor studentilor:

Tabela Courses Aceasta tabela contine informatii detaliate despre fiecare curs disponibil:

- **id** (PK) – identificator unic al cursului
- **title** – titlul cursului
- **description** – descrierea cursului
- **field** – domeniul de studiu
- **startDate** – data de incepere
- **endDate** – data de finalizare
- **sessions** – numarul de sesiuni
- **image** – imagine reprezentativa
- **slots** – locuri disponibile
- **language** – limba in care este predat cursul
- **price** – pretul cursului
- **teacher** – numele profesorului

Tabela Discount Aceasta tabela este folosita pentru gestionarea reducerilor aplicate cursurilor:

- **id** (PK) – identificator unic al reducerii
- **course_id** (FK) – referinta catre cursul asociat
- **percentage** – procentul de reducere

Tabela Enrollments Tabela care tine evidenta inscrierilor studentilor la cursuri:

- **id** (PK) – identificator unic al inscrierii
- **course_id** (FK) – referinta catre cursul la care s-a facut inscrierea
- **student_id** (FK) – referinta catre utilizatorul/studentul inscris

Tabela UserRoles Aceasta tabela defineste rolurile suplimentare asociate utilizatorilor:

- **id** (PK) – identificator unic
- **user_id** (FK) – referinta catre utilizator
- **role** – rolul specific (ex: student, teacher)

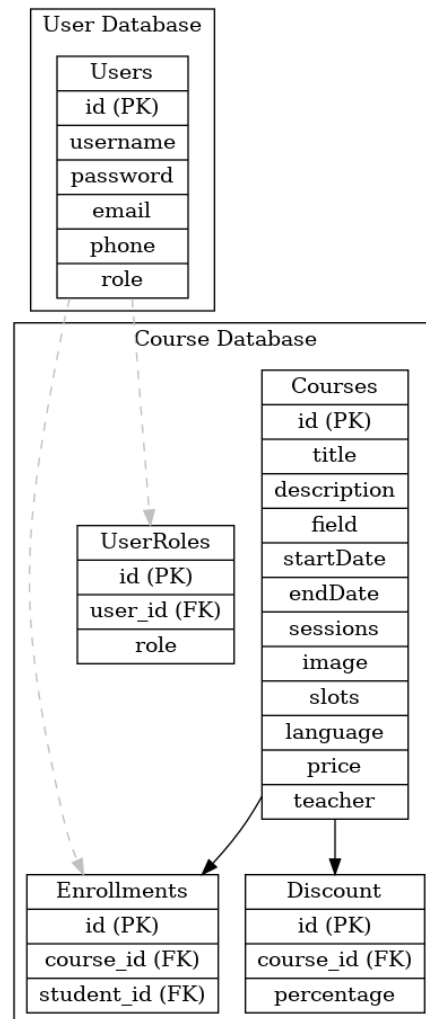


Figure 3: Modelul de date impartit pe doua baze de date

4 Interfete de comunicare

Sistemul este impartit in doua microservicii: **User Service** si **Course Service**. Fiecare microserviciu gestioneaza operatii asupra tabelelor din propria baza de date, prin API-uri REST.

4.1 User Service

User Service gestioneaza operatiile legate de utilizatori si autentificare. Endpoint-urile disponibile sunt:

- POST `/users` – creaza un utilizator nou
- GET `/users/:username` – returneaza datele unui utilizator dupa username
- PUT `/users/:username` – actualizeaza datele unui utilizator dupa username
- DELETE `/users/:username` – sterge un utilizator dupa username
- POST `/login` – autentificare utilizator (login)

4.2 Course Service

Course Service gestioneaza cursurile, inscrierile si reducerile, precum si functionalitatile de cautare.

Endpoint-uri pentru gestionarea cursurilor:

- POST `/courses` – creaza un curs nou
- GET `/courses/:title` – returneaza detalii despre un curs dupa titlu
- PUT `/courses/:title` – actualizeaza un curs dupa titlu
- DELETE `/courses/:title` – sterge un curs dupa titlu

Endpoint-uri pentru gestionarea reducerilor:

- POST `/discounts` – adauga o reducere pentru un curs
- GET `/discounts/:courseId` – returneaza reducerea unui curs

Endpoint-uri pentru gestionarea inscrierilor:

- POST `/enrollments` – inregistreaza un student la un curs
- GET `/enrollments/:courseId` – returneaza lista de studenti inscrisi la un curs

Endpoint-uri pentru cautare cursuri:

- GET `/courses?startDate=&endDate=` – cauta cursuri intre doua date

- GET /courses?field= – cauta cursuri dupa domeniu
- GET /courses?title= – cauta cursuri dupa titlu

Responsabilitatile sunt distribuite astfel: **User Service** gestioneaza utilizatorii si autentificarea, iar **Course Service** gestioneaza cursurile, inscrierile si reducerile.

5 Mockup-uri ale componentelor de interactiune cu utilizatorul

Frontend-ul aplicatiei va fi implementat in baza unor mockup-uri predefinite, realizate in etapa de design UI/UX. Aceste mockup-uri stabilesc structura si stilul vizual al principalelor componente ale interfetei grafice. Ele sunt punctul de plecare pentru implementarea efectiva a elementelor din frontend si asigura coerenta in design-ul final.

5.1 Mockup pagina de login

Mockup-ul reprezinta pagina de autentificare unde utilizatorii isi introduc credentialele (username si parola) pentru a accesa platforma.

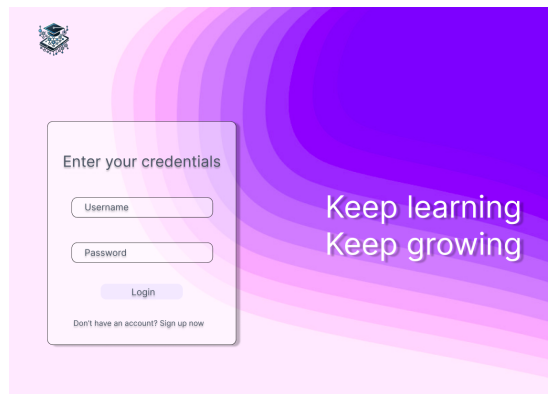


Figure 4: Mockup pagina de autentificare

5.2 Mockup pagina de register

Mockup-ul prezinta pagina de creare a unui cont nou, cu toate campurile necesare (username, parola, email, telefon) si optiunea de selectie a rolului (student sau profesor).



Figure 5: Mockup pagina de creare cont

5.3 Mockup pagina Home (utilizator)

Mockup-ul prezinta pagina Home vizibila pentru utilizatori care pot naviga pe platforma chiar si fara autentificare. In aceasta pagina, cursurile disponibile sunt afisate in grid si pot fi filtrate dupa criterii precum discount sau domeniu. Cursurile indisponibile sunt afisate cu font mai inchis, inasa nu pot fi apasabile.

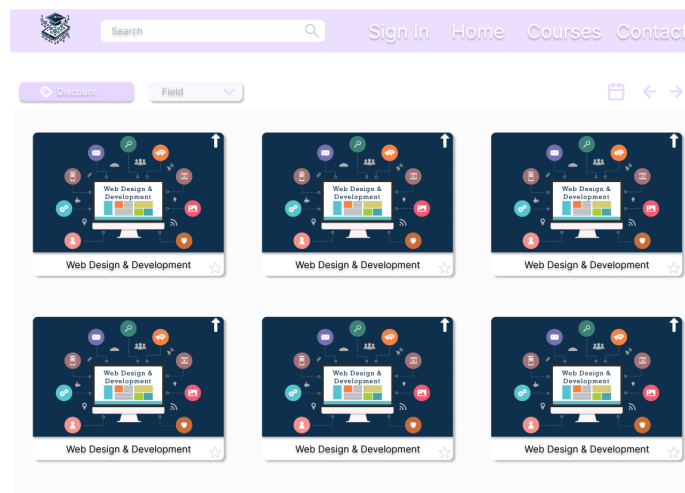


Figure 6: Mockup pagina Home - utilizator

5.4 Mockup pagina Home (profesor)

Mockup-ul prezinta interfata Home specifica pentru utilizatorii cu rol de profesor. Pe langa vizualizarea cursurilor, profesorul are acces la optiuni suplimentare de management al cursurilor (ex: editare sau stergere curs). Design-ul pastreaza

aceeasi structura de grid, insa elementele interactive sunt adaptate in functie de permisiuni.

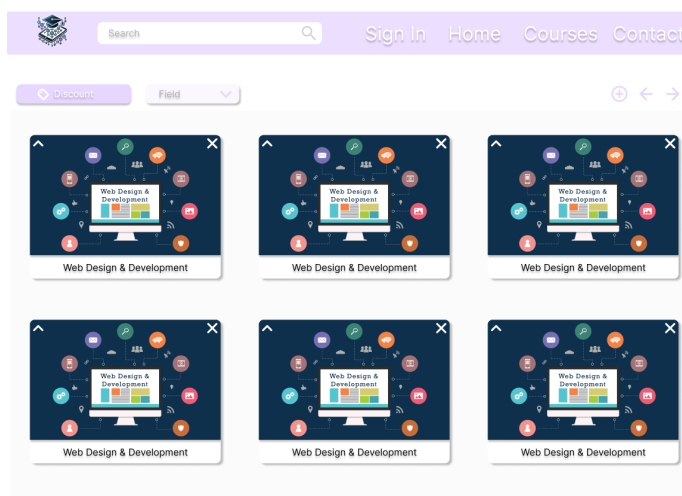


Figure 7: Mockup pagina Home - profesor

5.5 Mockup pagina de adaugare curs

Mockup-ul reprezinta pagina de adaugare a unui curs nou. Aceasta contine un formular complet cu campurile necesare: titlu, descriere, domeniu (field), data de inceput si de final, numarul de sesiuni, imagine asociata cursului, locuri disponibile, limba cursului si pretul. Pagina de Edit arata la fel, dar are si optiunea de a adauga un Discount pentru acel curs.

Figure 8: Mockup pagina de adaugare curs

5.6 Mockup pagina de detalii curs

Mockup-ul afiseaza detaliile complete ale unui curs selectat. Sunt vizibile imaginea cursului, titlul, numarul de sesiuni, perioada de desfasurare, limba in care este predat cursul, pretul si numarul de locuri disponibile. Sub acestea, utilizatorul poate consulta o descriere detaliata a cursului si are optiunea de a se inscrie prin butonul Enroll Now.

Figure 9: Mockup pagina de detalii curs

5.7 Mockup pagina de statistici

Mockup-ul prezinta pagina de statistici in care se vizualizeaza numarul de studenti inscrisi lunar, afisat sub forma de grafic. De asemenea, pagina include un calendar pentru selectarea unei date si afisarea listei de studenti inscrisi in acea zi.

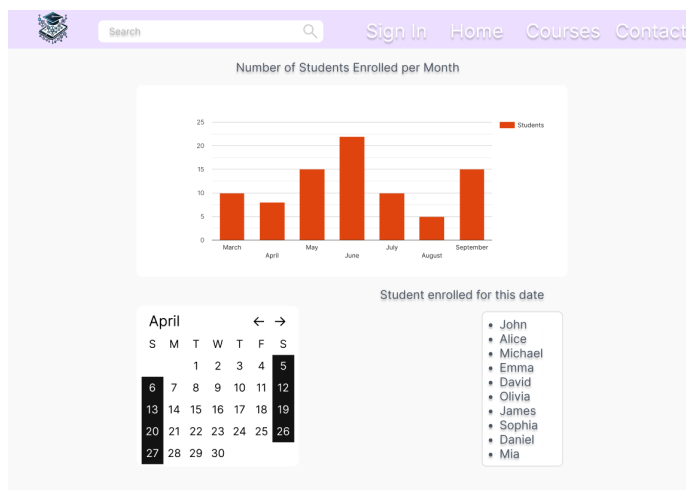


Figure 10: Mockup pagina de statistici studenti

5.8 Mockup pagina de contact cu chatbot activ

Mockup-ul afiseaza informatii de contact ale platformei, inclusiv adresa fizica, numarul de telefon, email-ul si programul de lucru. De asemenea, este vizibila si o componenta de tip chatbot in partea din stanga jos, care permite utilizatorului sa interactioneze cu asistentul virtual.

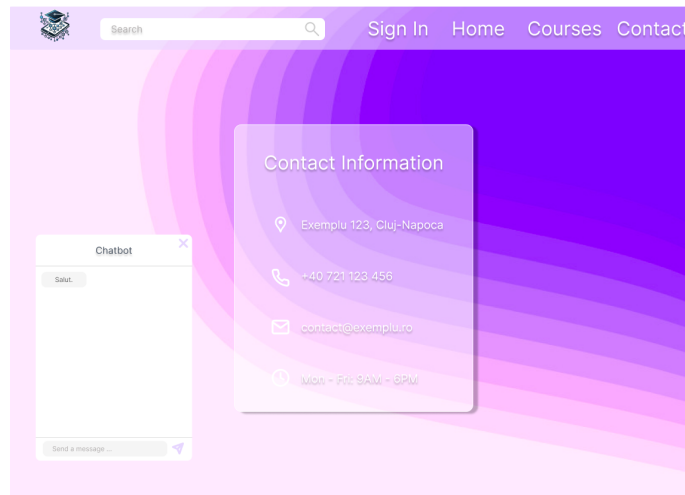


Figure 11: Mockup pagina de contact cu chatbot activ

5.9 Mockup pagina de contact cu chatbot inactiv

Mockup-ul prezinta aceeași pagina de contact, dar cu chatbot-ul în stare inactivă (minimizată), afișată sub forma unui icon în colțul din stanga jos.

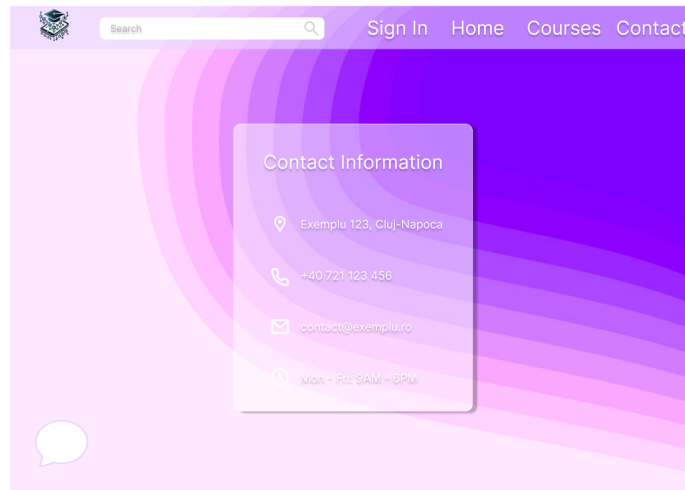


Figure 12: Mockup pagina de contact cu chatbot inactiv

5.10 Mockup pagina de profil

Mockup-ul reprezintă pagina de profil a utilizatorului, unde sunt afișate informații personale precum username-ul, parola, adresa de email și numărul de

telefon. Sub formularul principal, este prezent si un buton de **Sign Out** pentru deconectarea utilizatorului.

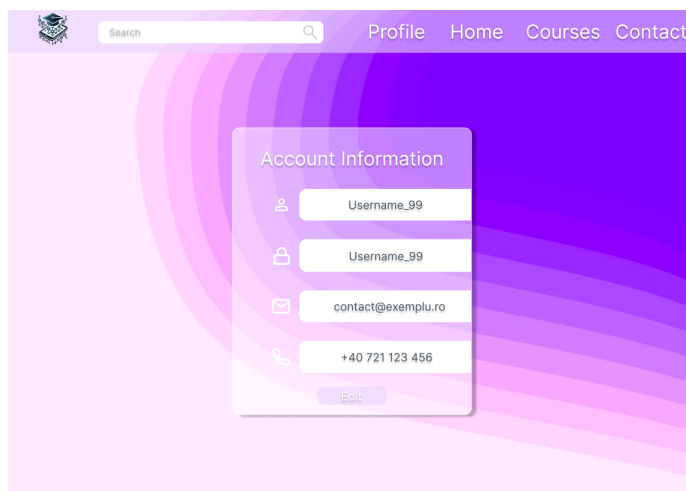


Figure 13: Mockup pagina de profil