

Programare WEB Learn Tribe

Proiectarea unei platforme de e-learning

Vlad Bartolomei

Grupa 30643

Profesor îndrumător: Mihnea Sebastian Burlacu



MEMBRĂ A



Facultatea de Automatică și Calculatoare
Universitatea Tehnică din Cluj-Napoca
România
Martie 2025

Cuprins

1	Arhitectură conceptuală	2
2	Arhitectura tehnică și tehnologii folosite	3
2.1	Angular	3
2.2	NestJS	3
2.3	MongoDB	3
3	Modelul de date	4
4	Interfețele de comunicare	4
5	Mockups	6
5.1	Vizitator neautentificat	6
5.2	Student	8
5.3	Profesor	9

Rezumat

Prezenta documentație își propune să descrie arhitectura conceptuală, să stabilească tehnologiile și să fie un îndrumător pentru curenții și viitorii dezvoltatori ai aplicației.

1 Arhitectură conceptuală

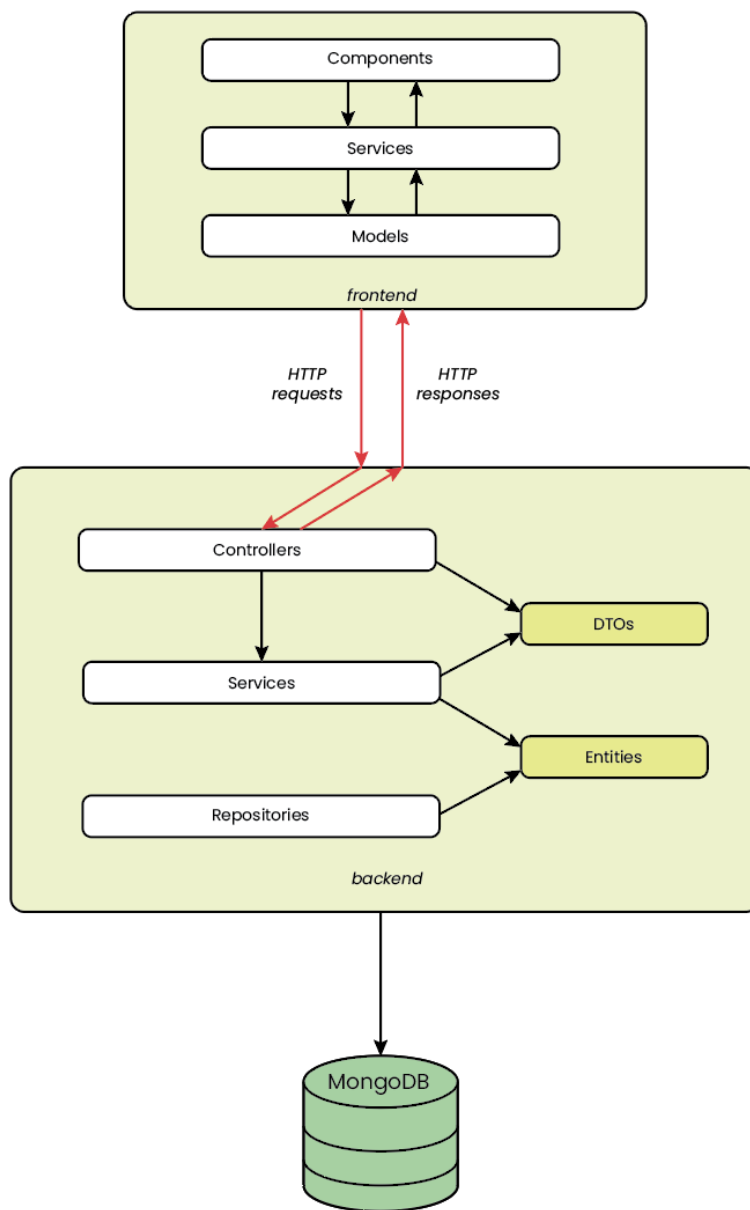


Figura 1: Arhitectura conceptuală

Pentru frontend avem arhitectura standard a oricărui proiect de Angular, constând în componente, servicii și modele (entități care vin de la backend). În proiectul nostru se vor regăsi următoarele pachete:

- 'core' - elemente esențiale, precum servicii, guards și interceptori HTTP;
- 'feature' - aici se vor regăsi componentele care descriu paginile aplicației. Fiecare componentă este unic determinată;
- 'shared' - vom regăsi atât clasele model, cât și unele componente care vor fi reciclate în proiect (spre exemplu, pagina de NotFound să poată fi apelată de orice componentă din 'feature'.

Pe partea de backend vom folosi această arhitectură Controller-Service-Repository, care transferă DTOs (Data Transfer Objects) între layer-uri până în Service, iar din Service are loc o mapare DTOs:Entities pentru a insera/persista obiectul în baza de date.

2 Arhitectura tehnică și tehnologii folosite

Pentru acest proiect vom folosi o arhitectură monolit.

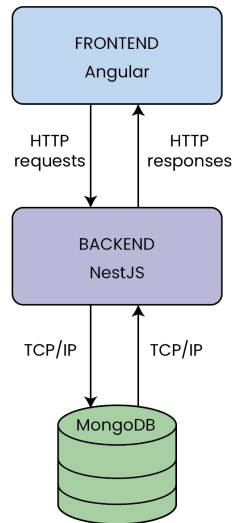


Figura 2: Stack-ul de tehnologii

2.1 Angular

Angular oferă un framework solid, modular și scalabil, potrivit pentru aplicații enterprise. Suportă RxJS pentru gestionarea fluxurilor de date reactive și permite utilizarea eficientă a componentelor reutilizabile.

Ca limbaj de scripting se folosește Typescript, puternic tipizat, însă care oferă și inferența tipului de dată. Acest aspect oferă o organizare mai clară și o lizibilitate mai bună a codului.

2.2 NestJS

NestJS este un framework bazat pe Express/Fastify, oferind un sistem modular, ușor de extins și menținut. Suportă integrarea cu MongoDB prin Mongoose și oferă suport nativ pentru WebSockets și microservicii.

Asemenea Angular, pentru scripting se folosește tot Typescript.

2.3 MongoDB

MongoDB este un sistem de baze de date NoSQL document-oriented, potrivit pentru aplicații ce necesită scalabilitate și flexibilitate ridicată. Mongoose permite definirea modelelor de date și validarea acestora într-un mod clar și structurat.

3 Modelul de date

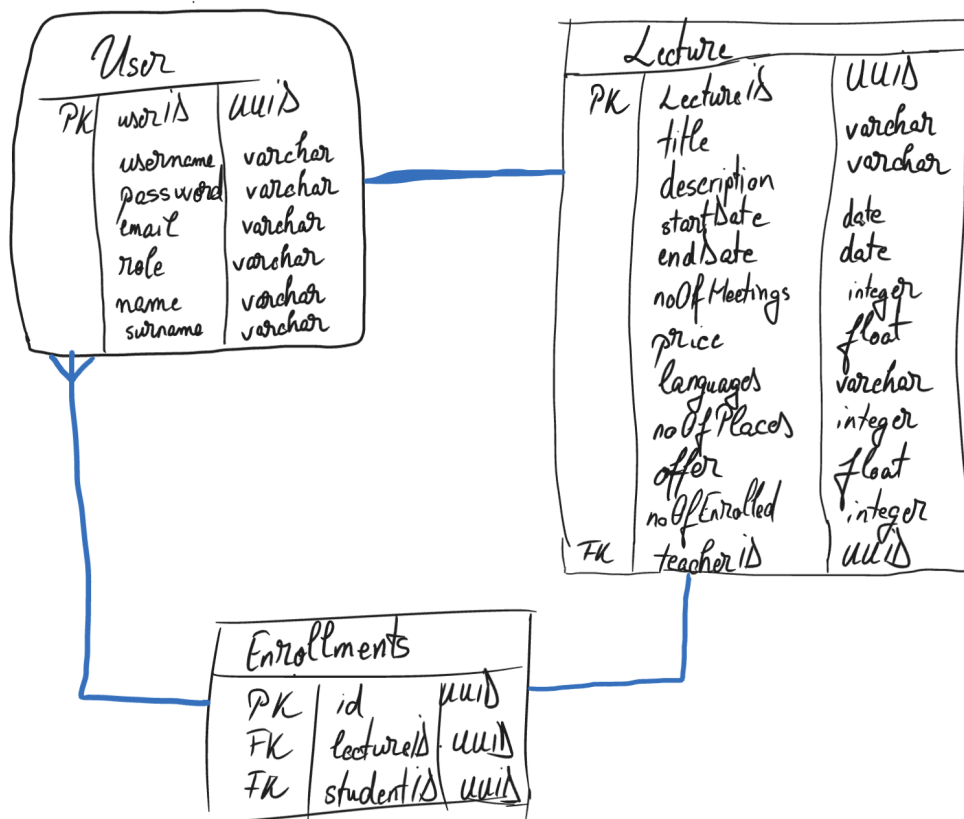


Figura 3: Modelul de date

4 Interfețele de comunicare

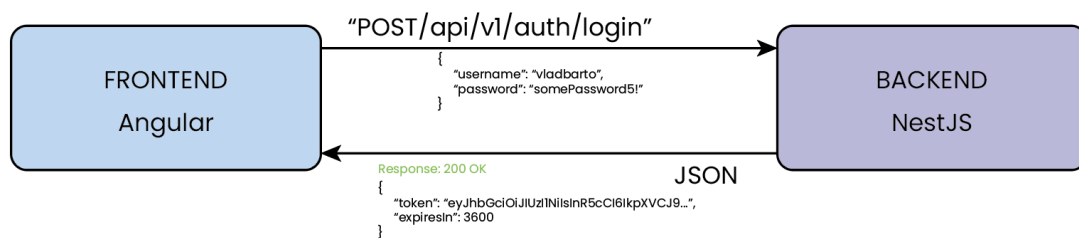


Figura 4: Scenariul de autentificare

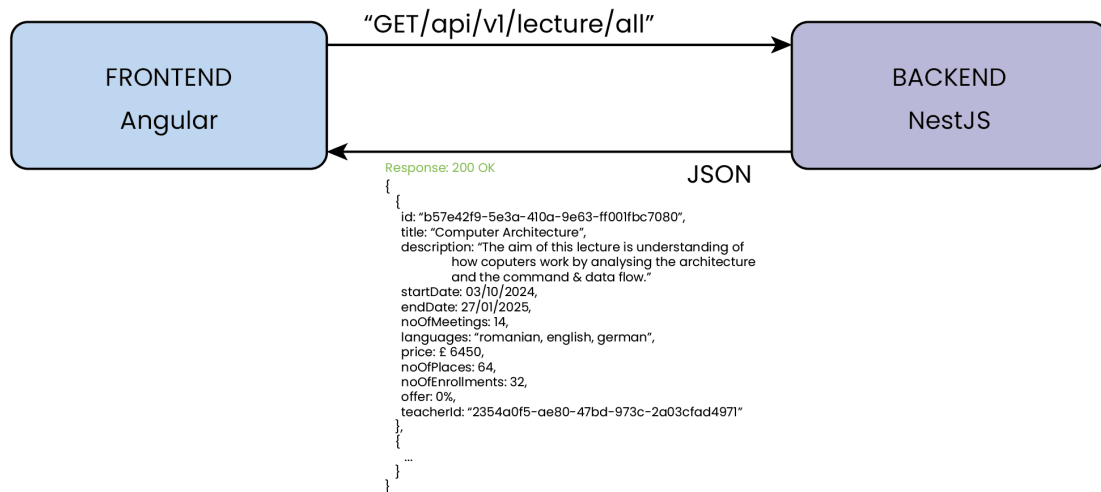


Figura 5: Aducerea tuturor cursurilor

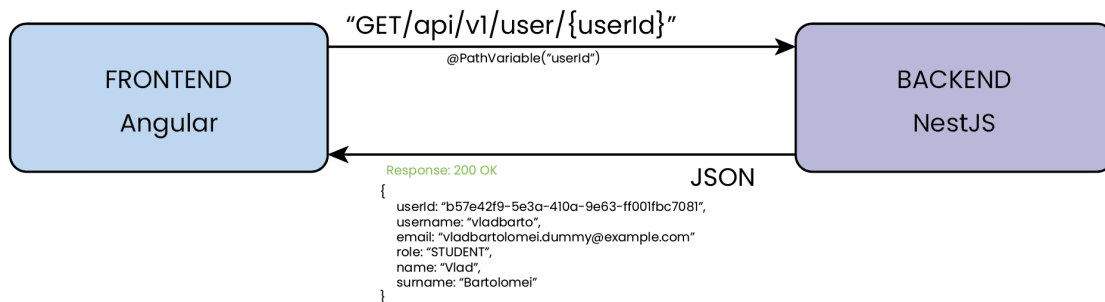


Figura 6: Accesarea informațiilor despre un utilizator

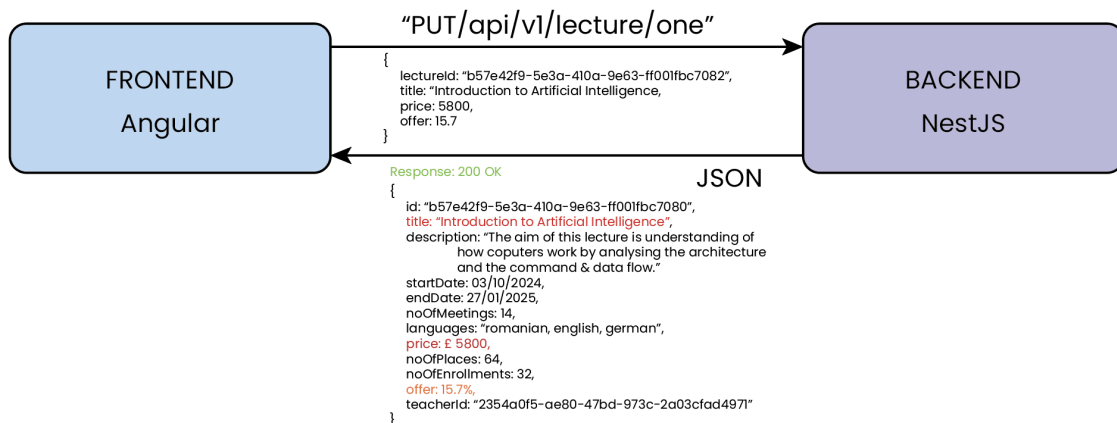


Figura 7: Actualizarea informațiilor unui curs

5 Mockups

În continuare vom prezenta mockups în diferite scenarii de utilizare, în funcție de starea de autentificare a utilizatorului.

5.1 Vizitator neautentificat

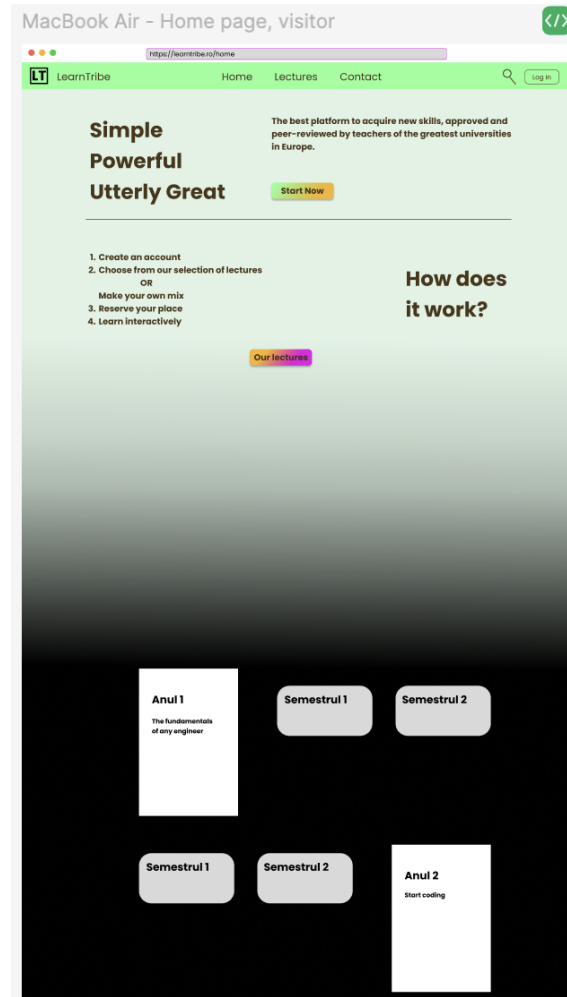


Figura 8: Pagina de acasă văzută de un utilizator neautentificat

Butonul de Start Now, în acest caz, redirecționează utilizatorul automat și logic către pagina de autentificare/înregistrare.

Butonul de Our Courses pornește o animație care derulează (scrolling) pagina curentă în jos, până la zona cu fundal cu intensitate redusă de culoare.

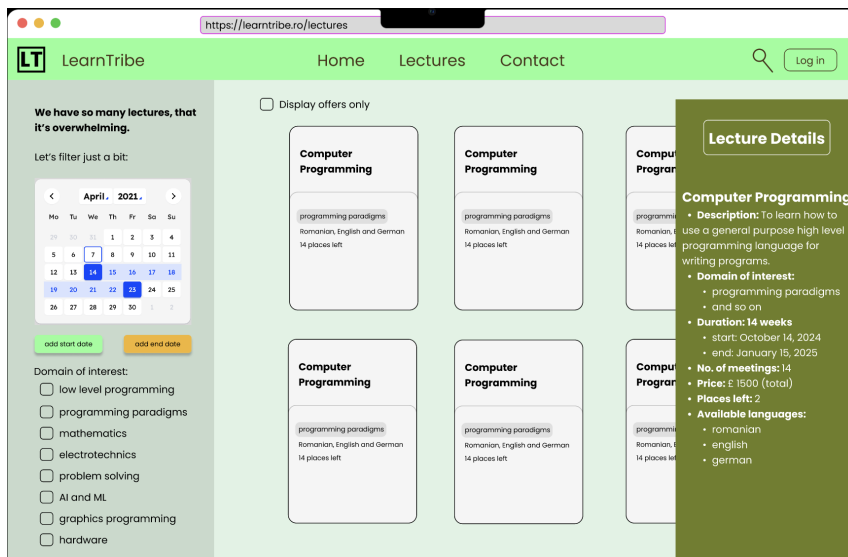


Figura 9: Pagina cu cursuri văzută de un utilizator neautentificat

Panoul lateral *Lecture Details* apare când se dă click pe unul dintre card-uri. Diferite filtre pot fi aplicate a.î. să filtreze rezultatele date. În mod ideal, rezultatele vor fi **paginate**.

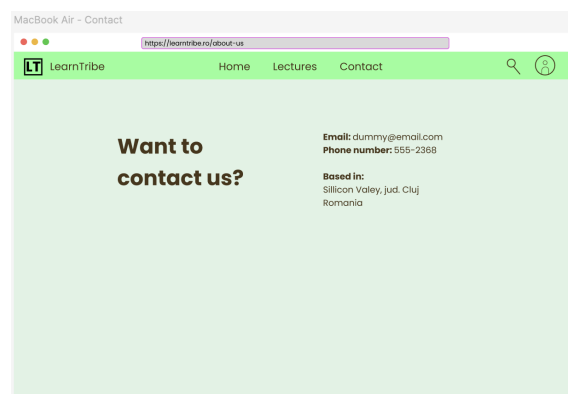


Figura 10: Pagina de contact, accesibilă oricărui utilizator, indiferent de starea de autentificare

Ce este specific acestor pagini este butonul de Login (dreapta sus).

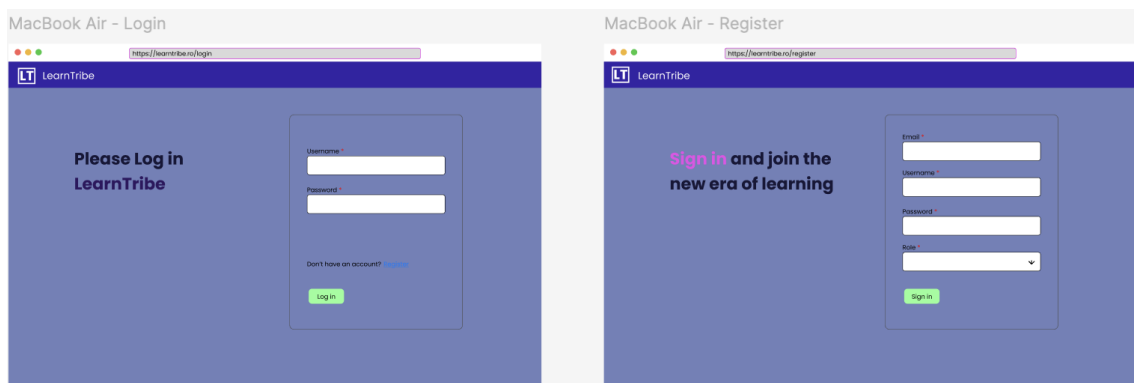


Figura 11: Autentificare sau înregistrare

Dropdown-ul de rol va permite două posibilități: STUDENT și PROFESOR.

5.2 Student

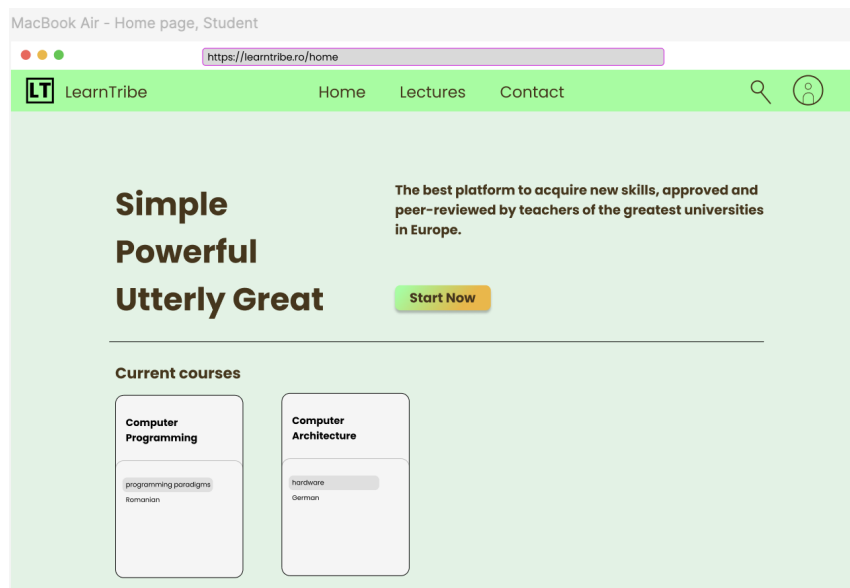


Figura 12: Pagina de acasă văzută de un utilizator autentificat ca student

Prima noutate aici este că studentul poate vedea cursurile curente la care este înrolat.

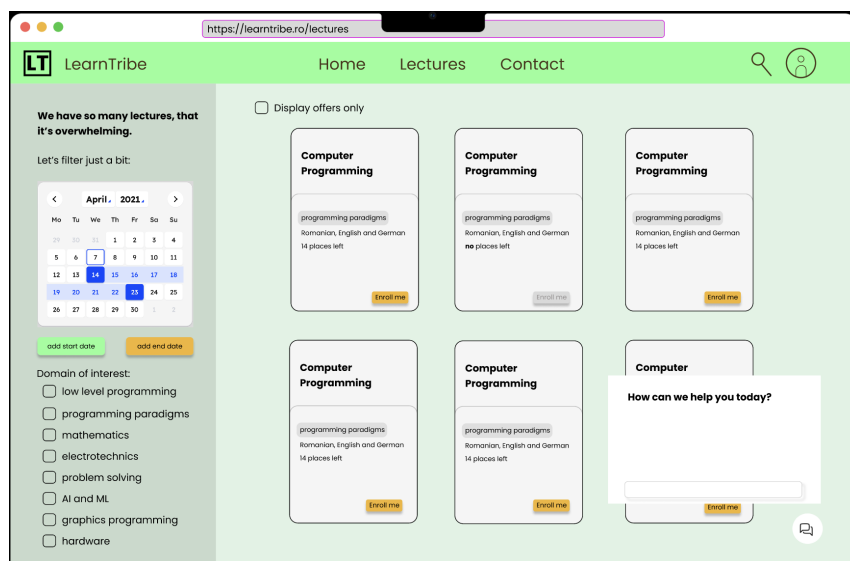


Figura 13: Pagina de cursuri văzută de un utilizator autentificat ca student

Două noutăți: interfața de chat cu chatbot-ul și butonul de Enroll Me. Dacă nu mai sunt locuri la cursuri, butonul este grayed out. Un caz ilustrat aici este cel în care studentul este deja înregistrat la un curs; se va afișa un mesaj de **Joined** pe card.

5.3 Profesor

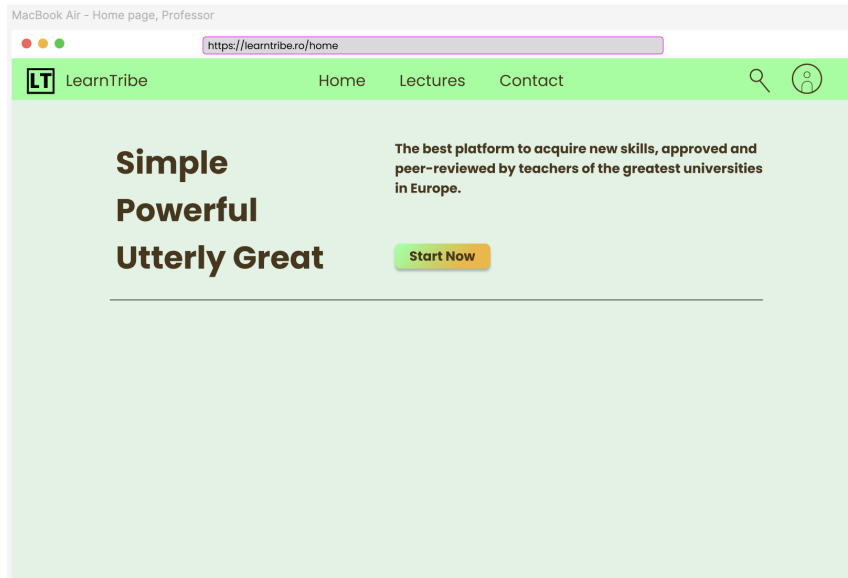


Figura 14: Pagina de acasă văzută de un utilizator autentificat ca profesor

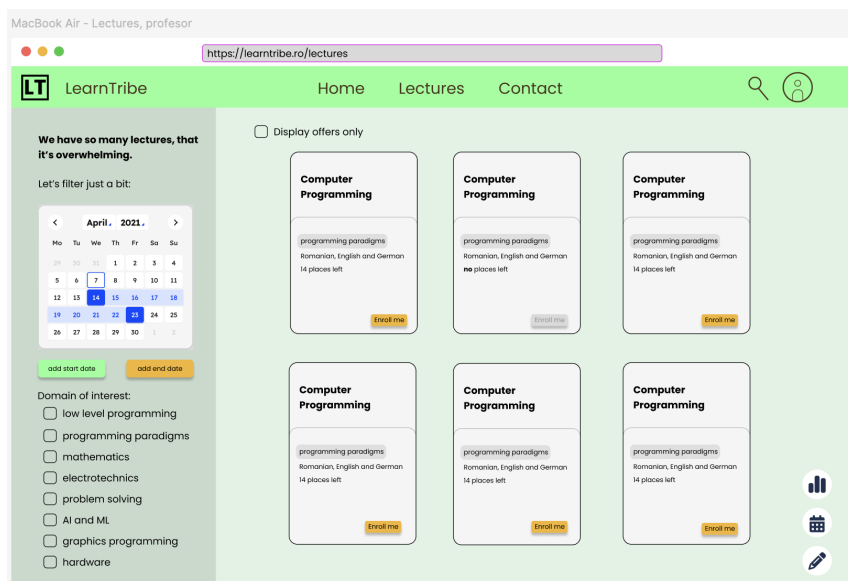


Figura 15: Pagina de cursuri văzută de un utilizator autentificat ca profesor

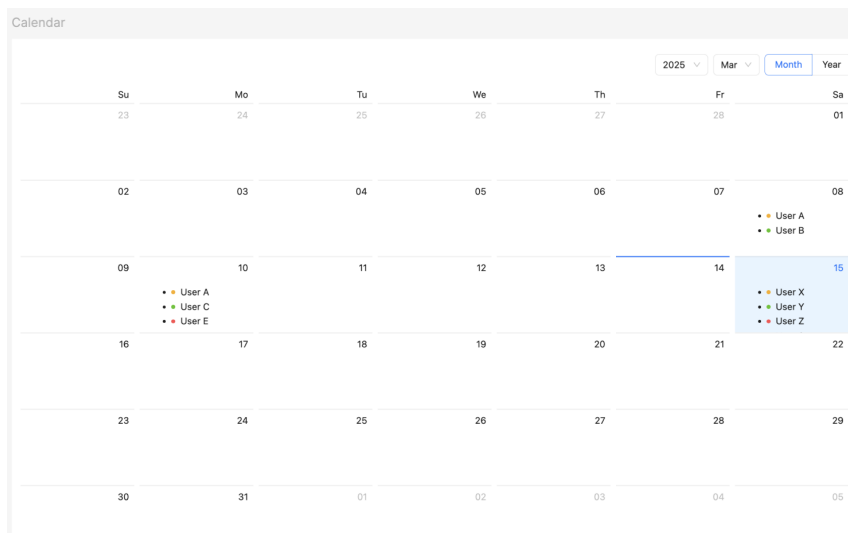


Figura 16: Calendar

Pe fiecare zi există o listă cu studenți înrolați. Dacă sunt prea mulți studenți, acea listă e scrollable.

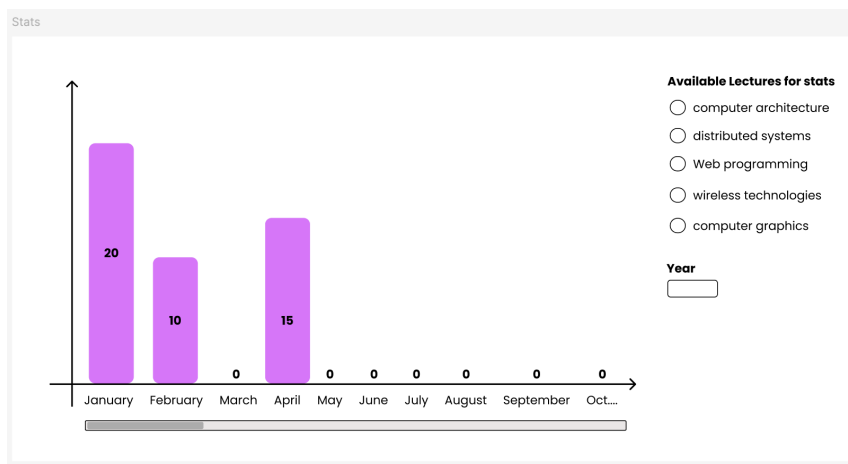


Figura 17: Statistici