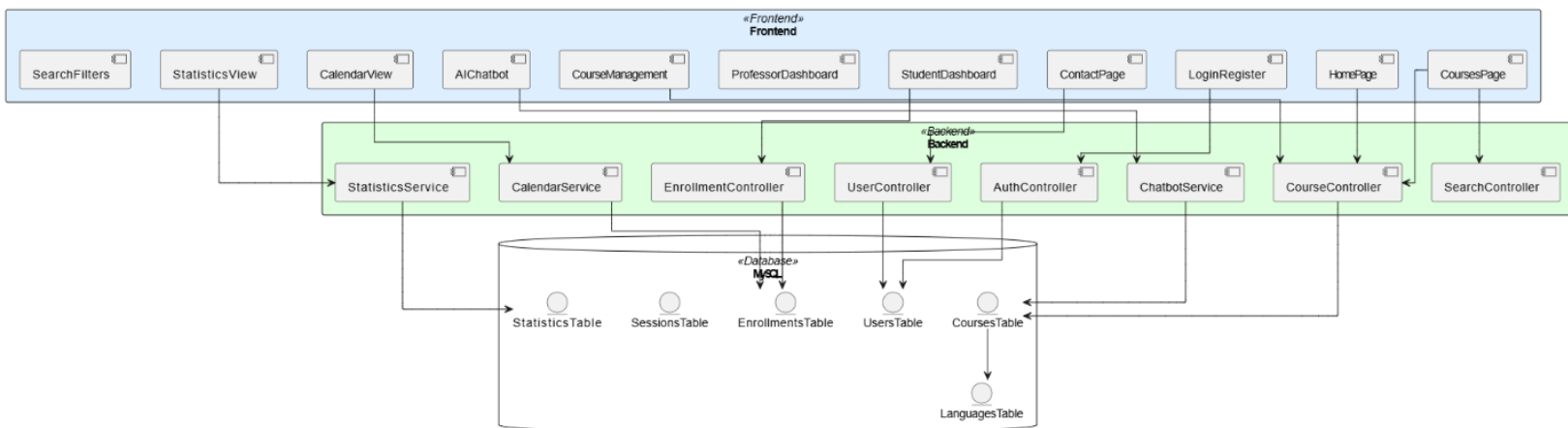
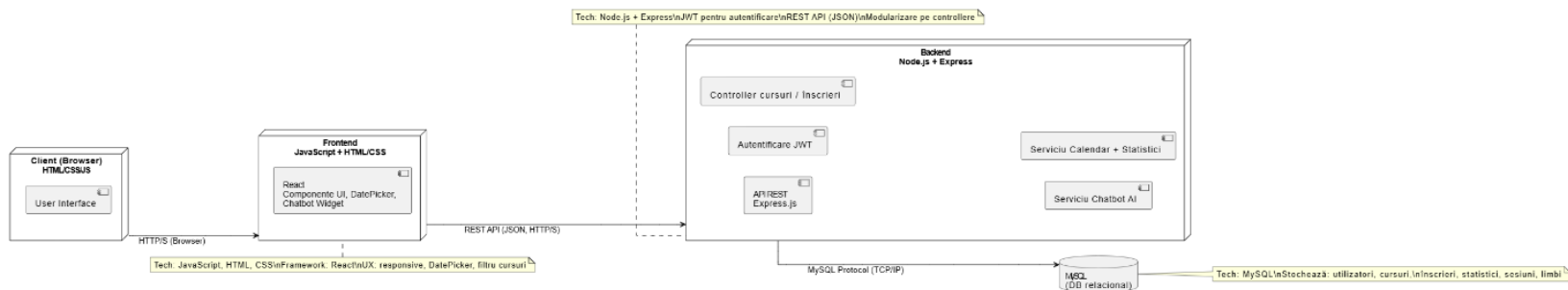


## Diagrama conceptuala



## Arhitectura tehnica



Platforma este organizată pe baza unei arhitecturi multi-strat (multi-tier), cu separarea clară a responsabilităților între client, interfața de utilizator, logică de business și stocare a datelor. Tehnologiile alese sunt moderne, scalabile și potrivite pentru dezvoltarea unei aplicații web interactive.

### 1. Client (Browser)

Tehnologii: HTML, CSS, JavaScript

Rolul clientului este de a accesa aplicația printr-un browser, încărcând interfața web și interacționând cu componentele din frontend.

Comunicarea se face prin protocolul HTTPS, pentru a asigura securitatea datelor transmise între client și server.

## 2. Frontend (React)

Tehnologii: JavaScript, React, HTML, CSS

Responsabilități:

- Afișarea interfeței utilizator (componente UI)
- Gestionarea filtrării cursurilor (DatePicker, filtre după domeniu)
- Widget de chatbot AI (UI)

React a fost ales pentru:

- Modularitate și reutilizarea componentelor
- Performanță în aplicații SPA (Single Page Application)
- Ecosistem bogat (librării pentru DatePicker, validare formulare etc.)

## 3. Backend (Node.js + Express)

Tehnologii: Node.js, Express.js

Responsabilități:

- Expunerea unui REST API care gestionează cererile din frontend
- Autentificare utilizatori prin JWT (JSON Web Tokens)
- Gestionarea cursurilor și înscrierilor prin controllere separate
- Serviciu Chatbot AI
- Generarea de date pentru calendar și statistici participare

Justificare alegeri:

- Node.js oferă performanță ridicată și este non-blocking, ideal pentru aplicații web scalabile

- Express permite o structurare clară pe rute și controllere
- JWT este o soluție modernă și sigură pentru autentificare

#### 4. Bază de date (MySQL)

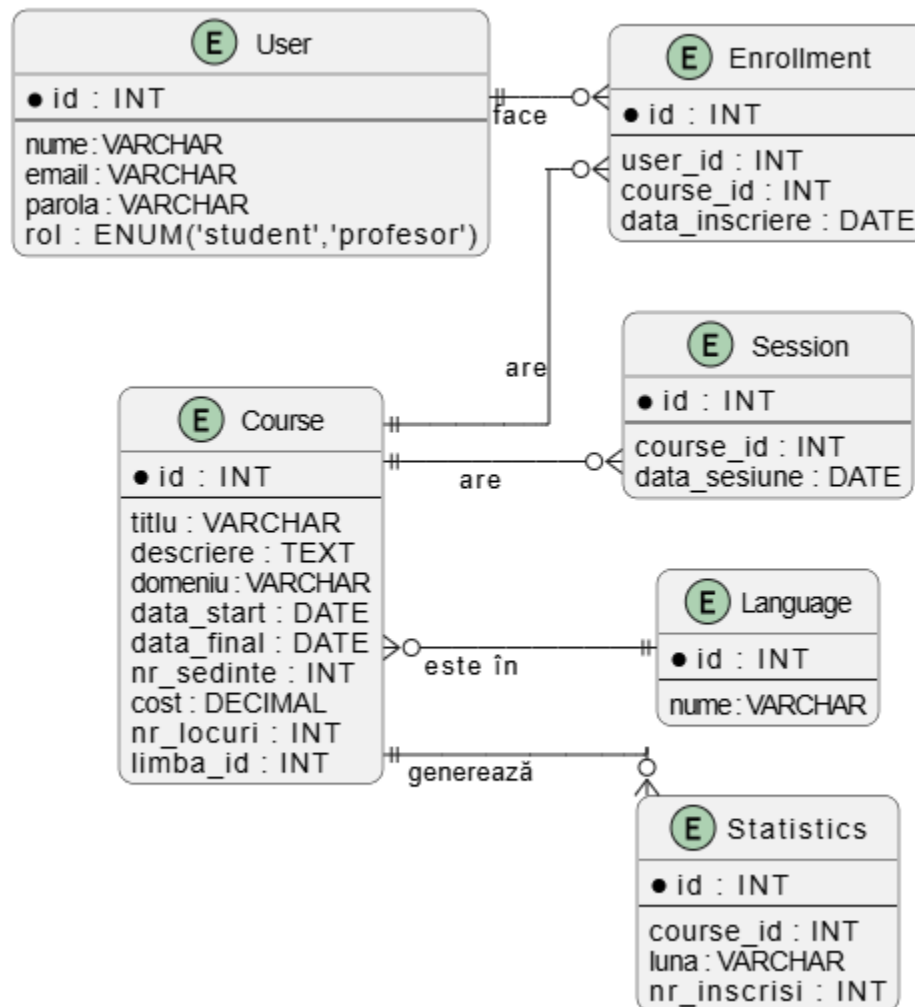
Tehnologie: MySQL (relațională)

Stochează: utilizatori, cursuri, înscrieri, statistici, sesiuni, limbi disponibile

MySQL a fost ales pentru:

- Structura clară și relațiile între entități
- Stabilitate și suport larg
- Interogări complexe (ex: filtrare cursuri, statistici participare)

## Model de date



## Interfete de comunicare

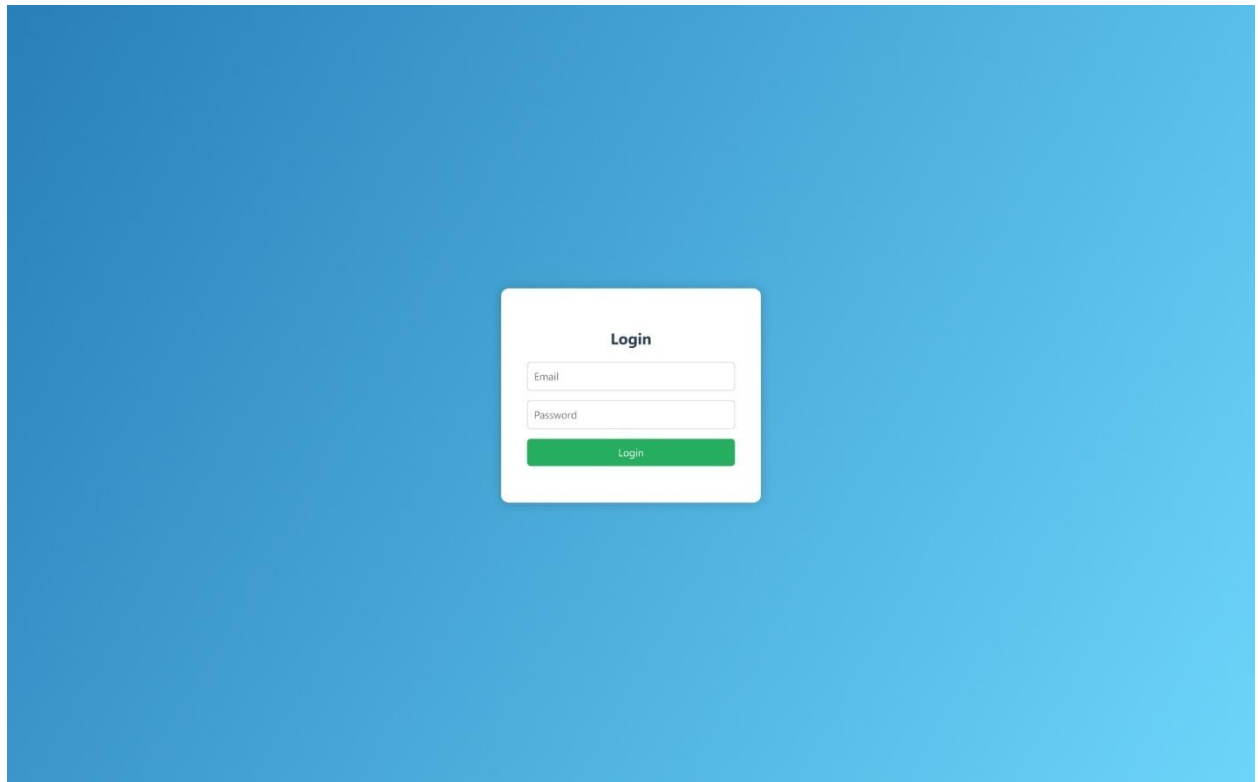
Backend-ul va expune o serie de API-uri REST, care vor permite frontend-ului să interacționeze cu datele din sistem, folosind formatul JSON și protocolul HTTPS.

| Method | Endpoint        | Description                        | Request Body / Params                       |
|--------|-----------------|------------------------------------|---|
| POST   | /api/auth/login | Authenticate user                  | { email, password }                         |
| GET    | /api/auth/info  | Get current user info              | Header:<br>Authorization:<br>Bearer <token> |
| GET    | /api/courses    | List courses with optional filters | -   |

|        |                                    |                                      |  |
|--------|------------------------------------|--------------------------------------|--|
| GET    | /api/courses/:id                   | Get course details                   | -  |
| POST   | /api/courses/add                   | Add a course<br>(professor)          | { title, description,<br>start_date, ... } |
| PUT    | /api/courses/update/:id            | Edit a course                        | { ... }                                    |
| DELETE | /api/courses/delete/:id            | Delete a course                      | -  |
| POST   | /api/enrollments/enroll            | Enroll student to a<br>course        | { course_id,<br>user_id }                  |
| GET    | /api/enrollments/getAllForUser/:id | List enrollments<br>for current user | -  |
| GET    | /api/enrollments/getByDate         | Enrollments by<br>date (calendar)    | ?course_id=1                               |
| GET    | /api/enrollments/course/get/:id    | Students enrolled<br>in a course     | -  |
| GET    | /api/statistics/courses/stats/:id  |                                      | Monthly enrollment stats                   |
| GET    | /api/statistics/allStats           |                                      | Stats for all courses                      |
| GET    | /api/sessions/getAll               |                                      | All sessions                               |
| POST   | /api/sessions/add                  |                                      | Add session (professor)                    |
| GET    | /api/sessions/get/:id              |                                      | Session details                            |
| GET    | /api/courses/languages/:id         |                                      | List available languages                   |
| POST   | /api/chatbot                       | Send question to<br>chatbot          | { question }                               |

# Mockups

## 1. Login Page

A mockup of a login page. The background is a solid blue color. In the center, there is a white rectangular box with rounded corners and a subtle drop shadow. Inside this box, the word "Login" is centered at the top in a bold, black font. Below the title, there are two input fields: the first is labeled "Email" and the second is labeled "Password", both in a small, gray font. At the bottom of the box, there is a green rectangular button with the word "Login" centered on it in a white font.

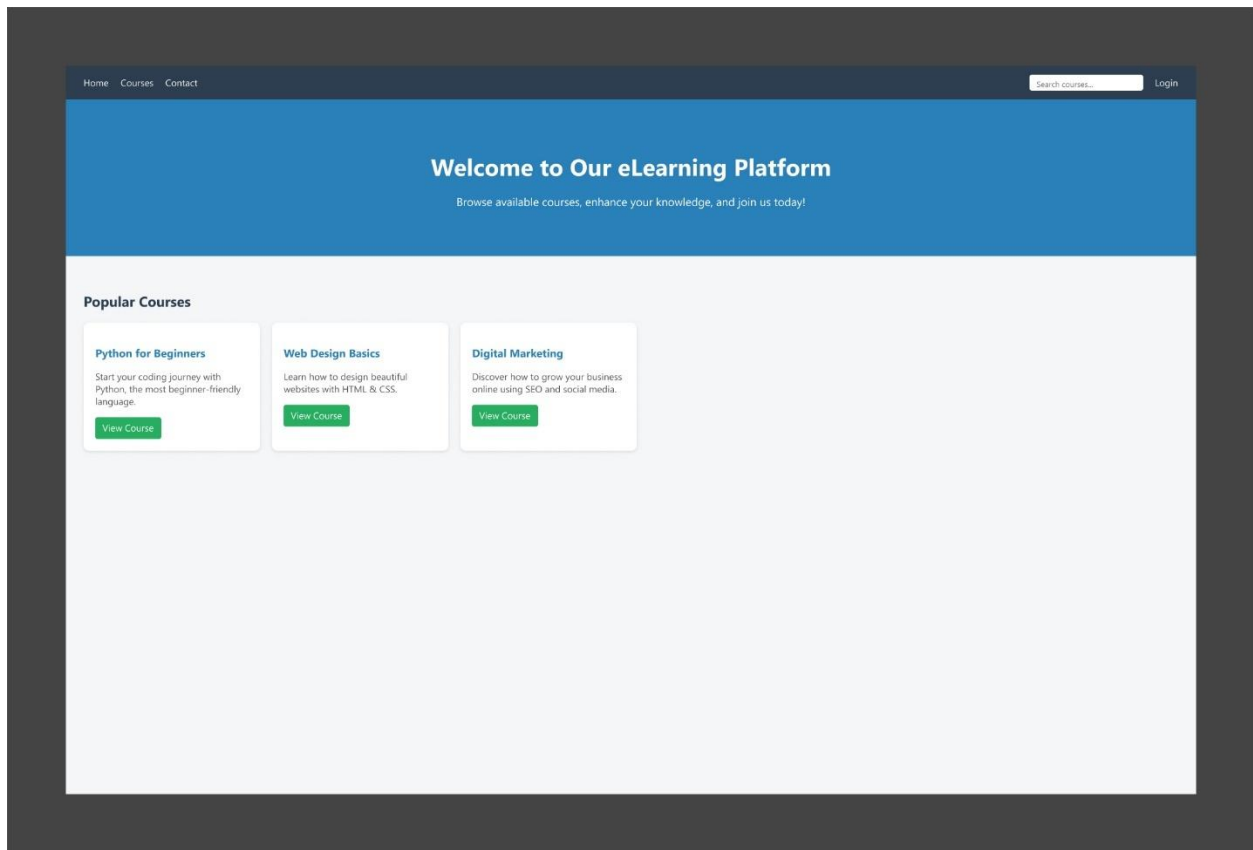
**Login**

Email

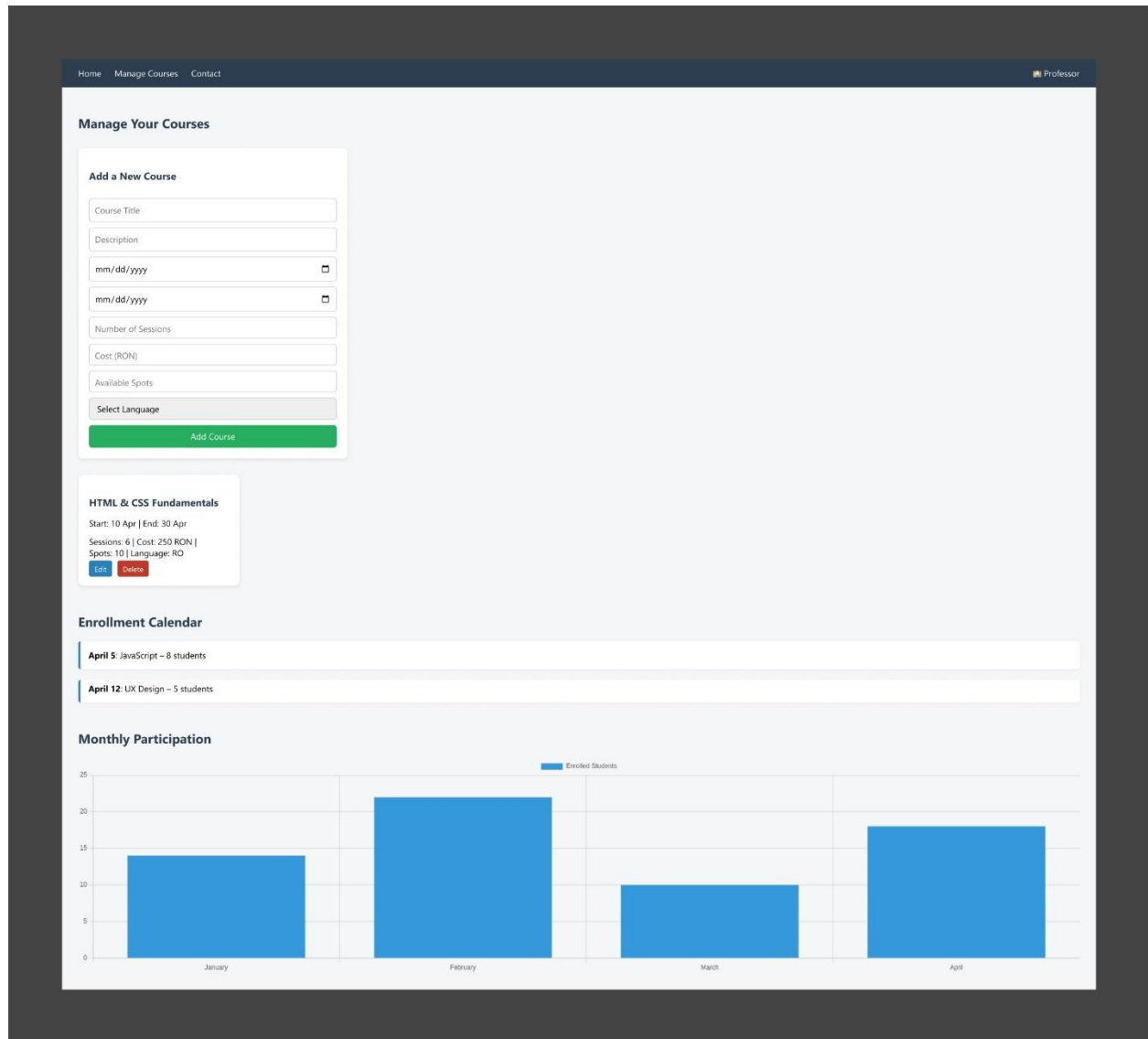
Password

Login

## 2. Home page for unregistered user



### 3. Course Page for Professor





## 4. Course Page for Student

[Home](#) [My Courses](#) [Contact](#) Student

### Available Courses

Start Date  
mm/dd/yyyy

End Date  
mm/dd/yyyy

Domain  
All

#### UX Design

Domain: Design

Dates: 1 Apr – 15 May

Sessions: 8 | Cost: 400 RON

Language: EN | Spots left: 0

Unavailable

#### JavaScript Advanced

Domain: Programming

Dates: 5 Apr – 30 May

Sessions: 10 | Cost: 500 RON

Language: RO | Spots left: 5

Enroll

[Need Help?](#)

## 5. Contact Page

