

A dark red background illustration featuring a smartphone on the left with a 'BET' button, a soccer ball in the center, and a soccer goal in the background. The words 'SPORTS BETTING' are faintly visible in the upper right.

**Aplicație de pariuri sportive**

**SPORTS-BETTING**

# Ideea proiectului

Aplicația este o platformă de pariuri sportive care oferă funcționalități distincte pentru trei tipuri de utilizatori: neînregistrați, înregistrați și administratori. Utilizatorii neînregistrați pot vizualiza meciurile și cotele, dar trebuie să se autentifice pentru a paria. Utilizatorii înregistrați pot paria individual sau în grupuri private, pot gestiona soldul (depuneri/retrageri), pot solicita autoexcluderea din platformă și au acces la istoricul biletelor și datele personale. Administratorii pot vizualiza activitatea tuturor utilizatorilor, edita sau dezactiva conturi, gestiona promoțiile, bloca cotele meciurilor, valida rezultate și aproba cereri de autoexcludere.

Platforma include un mod de pariere interactiv, în care utilizatorii pot crea grupuri private, pot contribui la bilete comune cu miză egală și împărțirea câștigurilor. Interfața conține pagini precum: Home (cu meciurile zilei, promoții și regulament), Login, Balanță, Biletele Mele și Gestionare Cont, iar meciurile sunt organizate pe competiții. Autentificarea este realizată cu JWT, iar aplicația oferă o experiență intuitivă și sigură pentru utilizatori

# Arhitectura aplicației

Aplicația este construită pe o arhitectură de tip client-server, având două componente majore:

- **Frontend:** React + TypeScript, responsabil pentru interfața cu utilizatorul, navigație și interacțiune (plasare pariuri, autentificare etc.). Comunicarea cu backendul se face prin cereri HTTP folosind Axios.
- **Backend:** Java Spring Boot, care implementează o arhitectură pe straturi:
  - Controller – primește cereri REST și trimite răspunsuri.
  - Service – conține logica de business (plasare pariuri, verificări, sold etc.).
  - Repository – accesează baza de date PostgreSQL prin JPA.
  - Model/Entity – definește structura datelor.
- **Autentificarea se face cu JWT (JSON Web Tokens):** utilizatorii primesc un token la login, care este trimis în toate cererile ulterioare și validat pe server. Accesul la anumite funcționalități este controlat prin roluri (Admin, Utilizator înregistrat, Neînregistrat).
- Meciurile, pariurile, utilizatorii și grupurile sunt entitățile principale, gestionate printr-o bază de date relațională. Arhitectura permite o separare clară a responsabilităților, testabilitate și extindere ușoară.

# Aspecte tehnice

- Dezvoltarea modului **de pariere interactivă în grup**, o funcționalitate originală care presupune mize comune și împărțirea câștigului între utilizatori.
- Implementarea autentificării JWT de la zero, fără framework extern (am renunțat la Keycloak), incluzând generarea, validarea tokenului și filtrele de securitate.
- Gestionarea accesului diferențiat pe roluri (admin, utilizator înregistrat, neînregistrat), cu protejarea endpointurilor în backend și a rutelor în frontend.
- Organizarea aplicației backend în straturi (Controller → Service → Repository), care permite separare clară a responsabilităților și testare ușoară.
- Integrarea Swagger/OpenAPI pentru documentarea și testarea rapidă a API-ului, fără a depinde de frontend.
- Configurarea Axios în React pentru trimiterea automată a tokenului JWT și tratarea răspunsurilor cu erori (401/403).
- Utilizarea unui API live pentru obținerea cotelor sportive în timp real, cu parsare și mapare din date externe în format JSON.

# Ce mai avem de implementat?

~În principal frontend~

Team Board

View 1

+ New view

Filter by keyword or by field

Discard

Save

Bugs0

This item doesn't work properly

Todo6

This item hasn't been started

Draft

Pagina Profil (cu sold/balanță)

Draft

Pagina Principală (meciuri + opțiune de pariere)

Draft

Pagina Bilete (cu istoricul pariurilor)

Draft

Verificare finala informatii Baza de Date

Draft

Verificare finala detalii

Draft

Verificare finală UI/UX

In Progress3

This is actively being worked on

Draft

Implementare OpenAPI + Swagger pentru testare vizuală

Draft

Pagina Home (redirecționare către celelalte pagini)

Draft

Interfață pentru navigare între pagini (Menu/Home)

Testing5

This item is in testing

Draft

Seeding in Baza de Date si migrare

Draft

Configurare login și logout din frontend

Draft

Protejarea endpoint-urilor cu filtre JWT

Draft

Creare roluri: Admin, Utilizator înregistrat și neînregistrat

Draft

Integrare API frontend-backend (Axios + JWT)

Done8

This has been completed

Draft

Integrarea PostgreSQL in Spring Boot

Draft

Diagrama Baza de date

Draft

Implementare CRUD pentru fiecare tabel

Draft

Creare Repository-uri si Controllerelor

Draft

Frontend- react typescript (Setup ul initial)

Draft

API-uri pentru fiecare Controller

Draft

Implementare autentificare cu JWT

Draft

Integrare API backend cu frontend (Axios)