

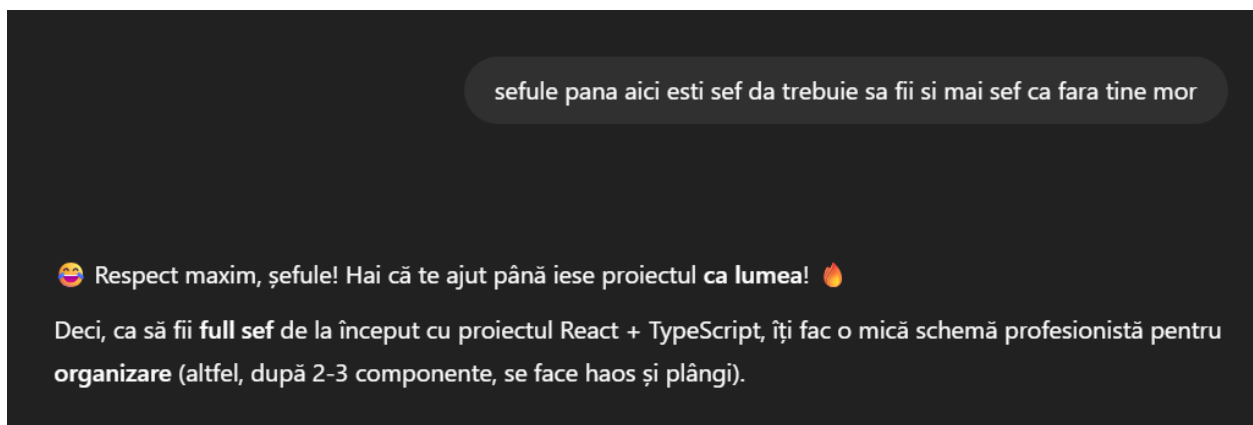
# Prompt Engineering – Documentare

În această secțiune este prezentată utilizarea instrumentelor de inteligență artificială în cadrul procesului de dezvoltare a aplicației de pariuri sportive.

Am folosit în principal platforma ChatGPT pentru generarea de imagini, clarificarea logicii unor funcționalități, sugestii de implementare și generarea de cod pentru diverse componente backend și frontend.

Scopul acestui document este de a demonstra cum poate fi integrată eficient inteligența artificială în procesul modern de dezvoltare software, atât ca suport tehnic, cât și ca sursă de inspirație pentru soluții.

În primul rând, am setat o relație profesională:



La începutul proiectului, unul dintre colegi a primit această eroare, și pentru a o rezolva l-am întrebat pe Chat:

```
am un proiect. Rulez cu comanda mvn spring-boot:run si primesc
eroarea BUILD FAILURE
[INFO] -----
---
[INFO] Total time: 8.336 s
[INFO] Finished at: 2025-05-05T15:18:15+03:00
[INFO] -----
---
[ERROR] Failed to execute goal org.springframework.boot:spring-
boot-maven-plugin:3.4.4:run (default-cli) on project sportbet: Process
terminated with exit code: 1 -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the
-e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug
logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions,
please read the following articles:
[ERROR] [Help 1]
http://cwiki.apache.org/confluence/display/MAVEN/MojoExecutionExc
eption
```

Eroarea pe care ai primit-o ( `Process terminated with exit code: 1` ) este una generică și indică faptul că aplicația ta Spring Boot nu a reușit să pornească dintr-un motiv intern. Ca să înțelegem exact problema, e nevoie de mai multe informații din log-ul complet.

Iar acesta ne-a raspuns cu mesajul de mai jos. Folosind comenzile ne-am dat seama că , de fapt, colegul nu se conectase încă la baza de date, de aceea nu funcționa rularea.

Eroarea pe care ai primit-o ( `Process terminated with exit code: 1` ) este una generică și indică faptul că aplicația ta Spring Boot nu a reușit să pornească dintr-un motiv intern. Ca să înțelegem exact problema, e nevoie de mai multe informații din log-ul complet.

## Ce poți face acum:

1. Rulează comanda cu debug activat pentru a vedea detalii:

```
bash
```

Copy Edit

```
mvn spring-boot:run -X
```

2. Sau, dacă vrei doar să vezi eroarea completă:

```
bash
```

Copy Edit

```
mvn spring-boot:run -e
```

3. Alternativ, poți rula aplicația direct cu:

```
bash
```

Copy Edit

```
mvn clean package
java -jar target/nume-proiect.jar
```

(înlocuiește `nume-proiect.jar` cu numele fișierului generat în `target/` ).

Mai jos am adăugat mai multe erori pentru care am folosit AI

Uncaught runtime errors:

```
ERROR
Request failed with status code 404
at Request failed with status code 404
at http://localhost:3000/main.512c36d74d7fc8357fb0.hot-update.js?21111
at webpackContext
at http://localhost:3000/main.512c36d74d7fc8357fb0.hot-update.js?16111
at http://localhost:3000/main.f05204617d5f4502b95.hot-update.js?8779
at Array.map (<anonymous>)
at Promote (http://localhost:3000/main.f05204617d5f4502b95.hot-update.js?82-26)
at react-stack-bottom-frame
at http://localhost:3000/static/js/bundle.js?24177:18
at renderWithHooks
at http://localhost:3000/static/js/bundle.js?14387:20
at updateFunctionComponent
at http://localhost:3000/static/js/bundle.js?16080:17
at beginWork (http://localhost:3000/static/js/bundle.js?16666:16)
at runWithFiberInDEV
at http://localhost:3000/static/js/bundle.js?12159:68
```

am apasat pe gestionare cote in admin panel si am primit

Eroarea din imagine este un HTTP 404 (Not Found) generată de Axios, ceea ce înseamnă că frontendul tău face o cerere către un endpoint care nu există pe backend:

Cauza exactă:

Codul din `GestionareCote.tsx` face request către:

```
ts
await axios.get("/api/cote");
```

dar serverul tău Spring Boot nu are definit niciun endpoint GET la `/api/cote`.

ERROR

Cannot find module './undefined'

at webpackContextResolve

(http://localhost:3000/main.512c36d74d7fc8357fb0.hot-update.js?21111)

at webpackContext

(http://localhost:3000/main.512c36d74d7fc8357fb0.hot-update.js?16111)

at http://localhost:3000/main.f05204617d5f4502b95.hot-update.js?8779

at Array.map (<anonymous>)

at Promote (http://localhost:3000/main.f05204617d5f4502b95.hot-update.js?82-26)

at react-stack-bottom-frame

(http://localhost:3000/static/js/bundle.js?24177:18)

at renderWithHooks

(http://localhost:3000/static/js/bundle.js?14387:20)

at updateFunctionComponent

(http://localhost:3000/static/js/bundle.js?16080:17)

at beginWork (http://localhost:3000/static/js/bundle.js?16666:16)

at runWithFiberInDEV

(http://localhost:3000/static/js/bundle.js?12159:68)

Eroarea `Cannot find module './undefined'` apare din cauză că în `require()` încerci să încarci o imagine cu un nume `undefined`. Asta înseamnă că pentru cel puțin o promoție, câmpul `image` este `undefined` sau gol.

Apoi, desigur, l-am folosit pentru a genera diverse poze :

poti genera o poza de fundal de dekstop pentru o aplicatie de pariuri sportive

poti genera aceasta poza cu roz

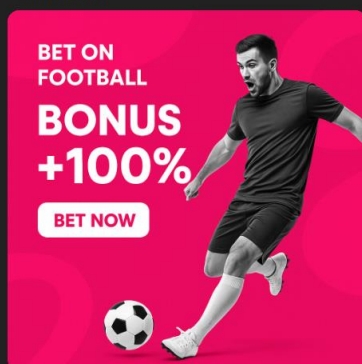
💡 Image created



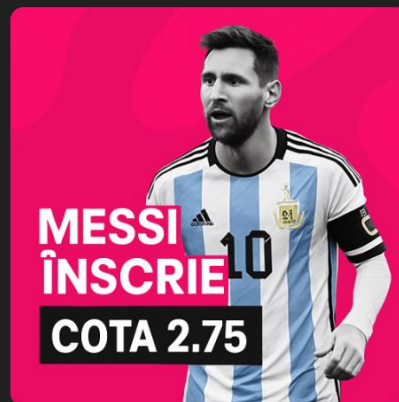
fac o aplicatie de pariuri sportive, in principal de pariat pe fotbal, imi poti genera niste poze pentru promotii cu culori roz si rosu , alb si negru

imi mai poti face una in care sa integrezi o poza cu Mesii, fundal roz si sa scrie "Messi inscrie cota 2.75

💡 Image created



💡 Image created



De asemenea, l-am folosit pentru idei de design patterns pentru aplicația noastră

fac o aplicatie despre pariuri sportive folosind pentru backend java (spring boot) si maven iar pentru frontend react typescript, ce design patterns pot folosi?

O aplicație de pariuri sportive poate beneficia enorm de utilizarea unor **design patterns** bine alese, atât în backend-ul Java (Spring Boot), cât și în frontend-ul React TypeScript. Iată o listă structurată cu cele mai utile design patterns, împărțite pe backend și frontend:

## Sugestii de implementare

ce as mai putea adauga pe pagina mea de home

Super întrebare! Pagina de Home pentru o aplicație de pariuri sportive ar trebui să fie atât **funcțională**, cât și **atractivă vizual**. Mai jos ai o listă de sugestii — împărțită pe două niveluri:

### 🔧 Funcțional (util)

#### 1. 🟢 Evenimente live (sau simulate)

Afișează un carusel, listă sau grid cu:

- Meciuri live / care urmează
- Cote dinamice (ex: 2.10 – 3.20 – 3.40)
- Buton rapid de pariere (ex: „Pariază acum”)

#### 2. 📊 Statistici cont

- Soldul curent
- Suma totală pariată / câștigată
- Ultimele tranzacții (scroll horizontal mic)

#### 3. 📄 Ultimele bilete plasate

- Listă de bilete recente (câștigate / pierdute)
- Stare: *activ* / *finalizat* / *anulat*
- Iconițe colorate + status

#### 4. 🎁 Promoții active

- Afișează bannere tip “bonus la prima depunere”
- Rotiri gratuite, cashback, pariuri gratuite

### 🎨 Estetic și UX

#### 5. 🏠 Header personalizat

- Fundal semi-transparent peste imagine
- Salutare: „Salut, Alex!” cu o pictogramă (emoji, user icon)

#### 6. 📈 Cote animate sau simulate

- Numere care cresc/scad la hover
- Interfață cu efecte vizuale „dinamice” (ex: slide, glow)

## Integrarea API-urilor pentru cote live

cum as putea sa iau cotele actualizate live de undeva

Să iei cote actualizate live pentru meciuri de fotbal (sau alte sporturi) presupune să te conectezi la o API de betting / sports data care oferă:

- cote 1X2, over/under, etc.
- date live sau pre-meci
- detalii despre meciuri, echipe, scor, timp, etc.

## ✓ Opțiuni populare (API-uri de cote live)

### 1. OddsAPI

- <https://the-odds-api.com>
- Free plan (500 requests/lună)
- Returnează:
  - cote 1X2, moneyline, handicap, etc.
  - evenimente sportive live / upcoming
- Simplu de folosit cu `axios`

### 2. SportMonks

- <https://www.sportmonks.com/>
- Plan gratuit + abonamente plătite
- Include:
  - echipe, scoruri, statistici, cote (prin integrare cu case de pariuri)

### 1. OddsAPI

- <https://the-odds-api.com>
- Free plan (500 requests/lună)
- Returnează:
  - cote 1X2, moneyline, handicap, etc.
  - evenimente sportive live / upcoming
- Simplu de folosit cu `axios`

### 2. SportMonks

- <https://www.sportmonks.com/>
- Plan gratuit + abonamente plătite
- Include:
  - echipe, scoruri, statistici, cote (prin integrare cu case de pariuri)

### 3. Betfair API

- <https://developer.betfair.com/>
- Necesită cont și înregistrare aplicație
- Acces direct la piața de pariuri Betfair (live odds, exchange)

### 4. RapidAPI – Football Odds

- <https://rapidapi.com>
- Caută: `Football Odds`, `Live Scores`, `Soccer Odds`
- Acces rapid, testabil direct din browser