



Academia de Studii Economice din București
Facultatea de Cibernetică, Statistică și Informatică Economică
Program de Masterat: SIMPRE

Dance Style Quiz

Coordonator:

Conf. Dr. Carmen Timofte

Student: Sima Andra-Cristiana

Grupa: 1134

Link video – prezentare proiect: https://youtu.be/zTMGr9kD1_Y?si=wYr0gCOJsiWnI07N

Link publicare: <https://dancequiz.vercel.app/>

Link GitHub: <https://github.com/AndraSima/Dance>

1. Introducere

Proiectul **Dance Style Quiz** este o aplicație web interactivă dezvoltată în React, care oferă utilizatorilor o recomandare personalizată de stil de dans pe baza unui quiz. Răspunsurile colectate sunt trimise către un model de limbaj AI (OpenAI ChatGPT), iar pe baza răspunsului generat, aplicația caută automat videoclipuri relevante pe YouTube folosind API-ul YouTube Data.

2. Descrierea problemei

Tot mai mulți oameni își doresc să înceapă să danseze atunci când dansul este văzut nu doar ca o formă de divertisment sau o formă de artă, ci și ca un mijloc eficient de relaxare, terapie și exprimare personală. Totuși, alegerea unui stil de dans potrivit este o provocare frecventă dacă nu ai un ghid personalizat sau o descriere clară a opțiunilor disponibile.

Deși mulți utilizatori se confruntă cu întrebări precum: „Mi s-ar potrivi mai bine un stil dinamic, precum hip-hop-ul, sau unul elegant, precum tango-ul?”, nu au un cadru accesibil în care să obțină o recomandare pe baza personalității, nivelului lor de energie sau preferințelor lor muzicale. Această alegere poate fi descurajantă sau aleatorie în lipsa unei îndrumări interactive.

Aplicația propusă vine în întâmpinarea acestei nevoi printr-o soluție inteligentă și accesibilă: un quiz interactiv care, prin analiza AI a răspunsurilor oferite, identifică stilul de dans care se potrivește cel mai bine fiecărui utilizator. În plus, aplicația facilitează învățarea și explorarea stilului sugerat prin furnizarea resurselor video relevante, care sunt automat extrase de pe YouTube. Prin urmare, problema este abordată nu numai prin furnizarea unui răspuns individualizat, ci și prin furnizarea de instrucțiuni precise pentru următorii pași, ceea ce face ca tranziția de la interes la acțiune să fie cât mai natural posibil.

3. Descriere API

a) OpenAI API (GPT-3.5 Turbo):

- Endpoint: <https://api.openai.com/v1/chat/completions>

- Metodă: POST

- Autentificare: Bearer Token

- Exemplu de utilizare:

```
prompt = `Pe baza acestor răspunsuri: ${answers.join(", ")}, ce stil de dans li s-ar potrivi...`  
axios.post("https://api.openai.com/v1/chat/completions",  
  { model: "gpt-3.5-turbo", messages: [...]} , { headers: { Authorization: `Bearer ${process.env.REACT_APP_OPENAI_KEY}` } })
```

b) YouTube Data API v3:

- Endpoint: <https://www.googleapis.com/youtube/v3/search>
- Metodă: GET
- Autentificare: API Key
- Exemplu de utilizare:

```
axios.get("https://www.googleapis.com/youtube/v3/search",  
  { params: { key: process.env.REACT_APP_YOUTUBE_KEY, q: `${style} dance tutorial`, part: "snippet", maxResults: 3 } })
```

4. Flux de date

Procesul de interacțiune cu Dance Style Quiz este structurat, ușor de înțeles și fluent, cu un flux de date clar definit între componentele aplicației și serviciile cloud utilizate (OpenAI și YouTube). Mai jos este o descriere detaliată a modului în care datele sunt transferate prin aplicație:

➤ Completarea quiz-ului (Quiz.js)

La deschiderea aplicației, utilizatorul este întâmpinat de o interfață prietenoasă, unde poate începe un quiz compus dintr-o serie de întrebări referitoare la stilul său de viață, preferințele muzicale, nivelul de energie și obiceiurile sale legate de activitate fizică. Aceste întrebări sunt definite într-un array de obiecte în fișierul Quiz.js, fiecare având opțiuni multiple de răspuns. Pe măsură ce utilizatorul răspunde, selecțiile sale sunt salvate într-un array local numit `answers`, prin funcția `setAnswers(newAnswers)`.

➤ Trimiterea datelor către OpenAI

Odată ce toate întrebările au fost completate, array-ul de răspunsuri (`answers`) este concatenat într-un string care este folosit drept prompt pentru un apel către API-ul OpenAI (modelul gpt-3.5-turbo). Acest prompt este trimis într-o cerere POST, cu un mesaj formulat astfel încât să solicite modelului identificarea unui singur stil de dans potrivit utilizatorului, în funcție de totalitatea răspunsurilor. Apelul este realizat prin intermediul bibliotecii Axios, cu autentificare pe bază de token stocat în variabila de mediu `REACT_APP_OPENAI_KEY`.

```
const response = await axios.post({
  "https://api.openai.com/v1/chat/completions",
  {
    model: "gpt-3.5-turbo",
    messages: [{ role: "user", content: prompt }],
  },
  {
    headers: {
      Authorization: `Bearer ${process.env.REACT_APP_OPENAI_KEY}`,
    },
  }
});
```

➤ **Primirea răspunsului de la OpenAI și stocarea rezultatului**

Răspunsul primit de la OpenAI este un text simplu (ex: „Salsa”), care este extras din obiectul **response.data.choices[0].message.content** și trimis mai departe către componenta principală a aplicației prin funcția **onResult(style)**. Aceasta setează starea result din componenta App.js cu stilul de dans recomandat.

➤ **Căutarea de videoclipuri YouTube (Result.js)**

Odată identificat stilul de dans, aplicația trece automat la componenta Result.js, unde stilul returnat este utilizat pentru a construi o interogare (query) către API-ul YouTube Data v3. Se generează o căutare automată de tipul "dance tutorial" (ex: "salsa dance tutorial"), iar rezultatele sunt obținute printr-un apel GET cu Axios. Răspunsul conține o listă de videoclipuri (obiecte items), din care se extrag titlurile și ID-urile pentru a fi afișate într-un player embedded YouTube.

```
const res = await axios.get(
  `https://www.googleapis.com/youtube/v3/search`,
  {
    params: {
      key: process.env.REACT_APP_YOUTUBE_KEY,
      q: `${style} dance tutorial`,
      part: "snippet",
      maxResults: 3,
    },
  }
);
```

➤ **Afișarea rezultatelor și posibilitatea de a relua testul**

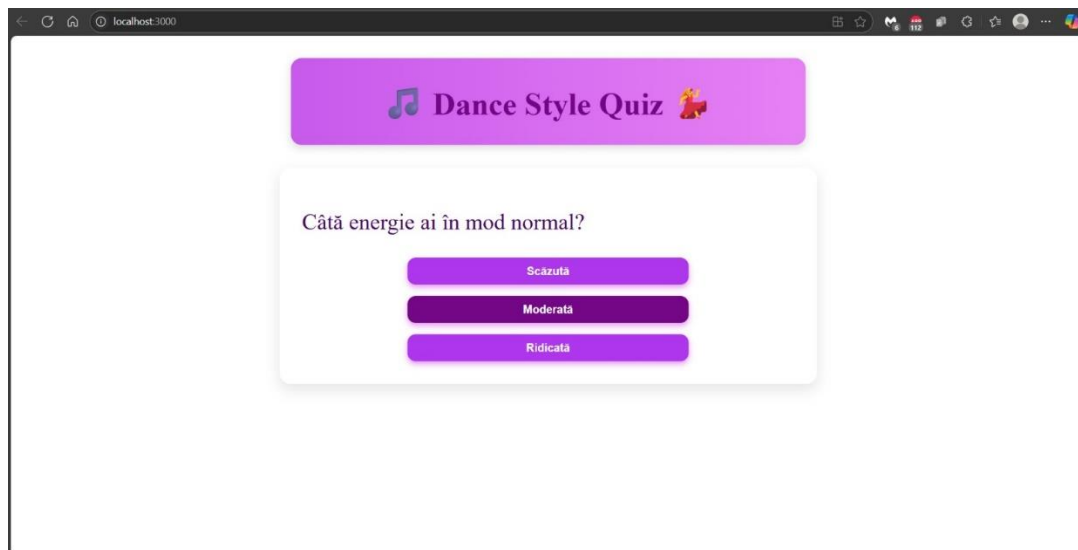
Utilizatorului i se afișează titlul stilului de dans recomandat și o serie de videoclipuri relevante, toate în cadrul unei interfețe atractive și responsive: **setVideos(res.data.items)**. La finalul paginii, există un buton „Reia testul”, care resetează starea aplicației (**setResult(null)**) și permite reluarea procesului de la început, în caz că utilizatorul dorește să obțină un rezultat diferit sau să reconsidere răspunsurile.

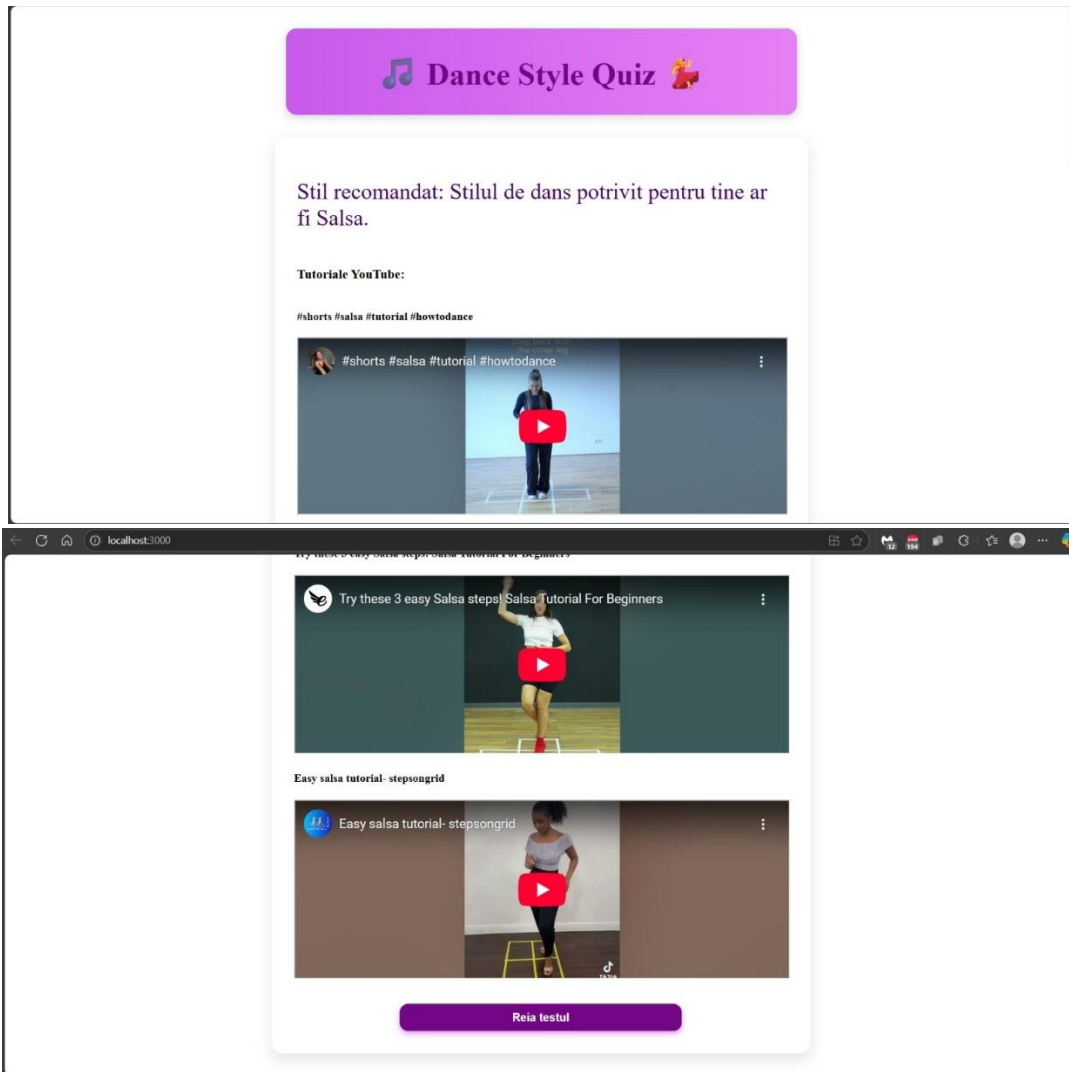
➤ **Autentificare și autorizare servicii utilizate:**

- OpenAI: Bearer Token în header-ul Authorization
- YouTube Data API: API Key în query string
- Datele de autentificare sunt ascunse în fișierul `.env` și accesate prin `process.env.REACT_APP_...`

Acest flux de date asigură o experiență fluidă, bazată pe procesarea inteligentă a informațiilor și livrarea de conținut personalizat, integrând perfect capacitățile a două servicii cloud majore: OpenAI (pentru analiză și decizie logică) și YouTube (pentru furnizarea de conținut multimedia adaptat contextului).

5. Capturi ecran aplicație





6. Referințe

<https://react.dev/learn>

<https://nextjs.org/docs>

<https://platform.openai.com/docs/concepts>

<https://learning.postman.com/docs/getting-started/first-steps/sending-the-first-request/>

<https://vercel.com/>