**ACADEMIA DE STUDII ECONOMICE DIN BUCUREȘTI**

*Facultatea de Cibernetică, Statistică și Informatică Economică*

**Crearea unei aplicaţii web care să utilizeze 2 servicii în cloud, prin intermediul unui API REST**

**Profesor**: Timofte Carmen **Studenți**:

Guriță Alexandru Bradu Adela

Cuprins

[Introducere 2](#_Toc134888352)

[Descriere problema 4](#_Toc134888353)

[Descriere API 5](#_Toc134888354)

[Flux de date 6](#_Toc134888355)

[Exemple de request / response 7](#_Toc134888356)

[Metode HTTP 8](#_Toc134888357)

[Capturi ecran aplicație 9](#_Toc134888358)

[Referințe 10](#_Toc134888359)

# Introducere

În ultimii ani, dezvoltarea de aplicaţii web a devenit din ce în ce mai populară datorită flexibilității, scalabilitații și ușurinței de utilizare. Un aspect important al dezvoltării de aplicații web moderne este utilizarea serviciilor în cloud și integrarea lor prin intermediul API-urilor REST.

În cadrul acestui proiect, ne vom concentra pe crearea unei aplicații web care să utilizeze două servicii în cloud diferite şi să le integreze prin intermediul unui API REST. Acest lucru ne va permite să demonstrăm abilitățile în ceea ce privește dezvoltarea de aplicații web, utilizarea serviciilor în cloud și integrarea prin intermediul API-urilor REST.

Aplicația web va fi creată într-un limbaj de programare web modern și va folosi două servicii în cloud diferite. Vom utiliza un API REST pentru a conecta cele două servicii și a permite utilizatorilor să interacționeze cu ele printr-o interfață web simplă și intuitivă.

În acest proiect vom descrie procesul de dezvoltare al aplicației web, vom discuta despre beneficiile utilizării serviciilor în cloud și a API-urilor REST, precum și despre provocările întâmpinate în acest proces. Vom evidenția, de asemenea, principalele caracteristici ale aplicației web.

Această aplicație este o platformă de gestionare a unei liste de filme și descrierilor lor, care utilizează serviciul MongoDB Atlas pentru stocarea și gestionarea datelor, împreună cu metodele HTTP GET, POST și DELETE pentru interacțiunea cu acestea. De asemenea, aplicația include și un asistent virtual, care este integrat prin intermediul OpenAI API Key și poate ajuta utilizatorii în căutarea de noi filme, sugestii și multe altele.

**Un mic video de prezentare poate fi văzut accesând următorul link: https://youtu.be/ionOfXLL\_PU, iar aplicația are deploy pe platforma Vercel, disponililă aici: cloud-computing-project-9jc3k1f63-adelabradu.vercel.app.**

Există mai multe beneficii ale utilizării serviciilor în cloud și a API-urilor REST, printre care se numără:

* Scalabilitate - Serviciile în cloud permit o scalabilitate ușoară și rapidă a aplicațiilor, deoarece resursele pot fi alocate sau dealocate în timp real, în funcție de necesitățile utilizatorilor. API-urile REST permit integrarea eficientă a acestor servicii și permit aplicației să crească și să se adapteze la cerințele utilizatorilor.
* Accesibilitate - Serviciile în cloud sunt accesibile de oriunde, deoarece nu este necesară instalarea de software sau hardware local, ceea ce face ca aplicațiile să fie disponibile în orice moment și în orice locație. De asemenea, API-urile REST permit integrarea aplicației cu alte servicii web sau aplicații, permițând utilizatorilor să acceseze și să utilizeze diverse resurse și informații de pe internet.
* Cost redus - Serviciile în cloud elimină necesitatea achiziționării și întreținerii infrastructurii hardware și software, ceea ce face ca numeroasele costuri de dezvoltare și întreținere a aplicațiilor să fie reduse semnificativ. De asemenea, API-urile REST sunt proiectate pentru a fi utilizate eficient, făcând posibilă integrarea simplă și ieftină a mai multor servicii și aplicații.
* Flexibilitate - Utilizarea serviciilor în cloud și a API-urilor REST permite dezvoltatorilor să utilizeze diverse platforme, limbaje de programare și tehnologii, fără a fi limitați de restricțiile hardware sau software locale. Această flexibilitate face posibilă dezvoltarea de aplicații inovatoare și scalabile.
* Securitate - Serviciile în cloud sunt protejate de măsuri de securitate avansate, iar dezvoltatorii pot beneficia de această securitate, fără a fi nevoie să aloce resurse suplimentare pentru securitatea propriilor aplicații. De asemenea, API-urile REST permit dezvoltatorilor să integreze diverse măsuri de securitate și autentificare pentru a proteja informațiile și resursele utilizatorilor.

# Descriere problema

În era digitală, ne-am obișnuit să avem acces la o varietate de filme și emisiuni TV de pe o mulțime de platforme online. Cu toate acestea, în căutarea unui film sau a unei serii (sau serial) pe care dorim să le vizionăm, putem ajunge să pierdem mult timp căutând și uitându-ne la trailere, citind recenzii și verificând disponibilitatea pe diferite site-uri. Dar ce-ar fi dacă ar exista o soluție simplă care să ne permită să adăugăm filmele și serialele pe care le dorim într-un singur loc, pentru a le viziona mai târziu? Aceasta este ideea din spatele aplicației web care îți permite să introduci filmele la care vrei să te uiți ulterior pe care am dezvoltat-o.

Această aplicație web îți permite să îți creezi propriul playlist de filme și seriale pe care dorești să le urmărești. Prin intermediul unei interfețe intuitive, poți adăuga filme în lista ta de vizionare alături de descrierile acestora, urmărindu-le în timp ce sunt afișate într-un format curat și organizat. Poți adăuga filme și seriale cu ușurință la listă pentru o accesibilitate mai ușoară.

Pe lângă lista personalizată de filme și seriale, aplicația poate oferi și alte funcții utile, cum ar fi posibilitatea de a întreba un asistent virtual configurat ca un specialist în filme direct în cadrul aplicației. De asemenea, îți poate oferi detalii despre disponibilitatea filmelor și serialelor pe diferite platforme de streaming, astfel încât să poți decide de unde să le vizualizezi. Totodată, poți cere păreri sau recomandări și sugestii pe baza întrebărilor pe care i le pui.

Această aplicație web este o soluție simplă și utilă pentru a îți organiza lista de filme și seriale pe care dorești să le vezi, oferindu-ți o modalitate ușoară de a adăuga, organiza și gestiona filmele preferate, având în vedere ca după ce le gestionezi, le poți șterge din lista ta.

# Descriere API

API (Application Programming Interface) reprezintă un set de protocoale, metode și instrumente de dezvoltare software care permit comunicarea și interacțiunea între diferite sisteme și aplicații. În mod general, API-ul reprezintă o interfață prin intermediul căreia două sau mai multe aplicații pot comunica și transmite date sau informații între ele.

API REST (Representational State Transfer) reprezintă un set de principii și convenții pentru dezvoltarea de API-uri, care permit comunicația între aplicații web și servere folosind protocolul HTTP/HTTPS. API-urile REST sunt bazate pe arhitectura client-server și sunt concepute pentru a fi scalabile, ușor de utilizat și interoperabile cu diferite platforme.

Principiile de bază ale API-urilor REST include:

* Utilizarea metodelor HTTP standard (GET, POST, PUT, DELETE etc.) pentru a defini operațiunile de bază pe resurse.
* Folosirea URI-urilor (Uniform Resource Identifiers) pentru a identifica și accesa resursele.
* Transmiterea datelor între client și server într-un format standard, cum ar fi JSON sau XML.
* Menținerea stării aplicației doar pe client, iar serverul să fie stateless (fără a menține informații despre starea aplicației).

API-urile REST sunt folosite în mod obișnuit pentru a permite accesul la resurse și servicii existente, cum ar fi datele din bazele de date sau informațiile disponibile prin intermediul altor API-uri web. API-urile REST sunt utilizate pe scară largă în dezvoltarea de aplicații mobile și web, și sunt considerate un standard în dezvoltarea de API-uri.

Există multe API-uri REST disponibile pentru cloud computing, iar acestea permit dezvoltatorilor să utilizeze serviciile cloud pentru a construi și rula aplicații scalabile și eficiente, iar în continuare vom descrie câteva din tehnologiile folosite.

ChatGPT este un model de limbaj natural, dezvoltat de OpenAI, care poate fi folosit pentru a genera răspunsuri la diverse întrebări și conversații. Acesta face parte din categoria modelelor de învățare profundă și a fost creat cu ajutorul arhitecturii GPT (Generative Pre-trained Transformer). ChatGPT este capabil să înțeleagă și să producă răspunsuri într-un mod similar cu modul în care o persoană ar putea comunica, oferind astfel o experiență de conversație naturală. De asemenea, acesta este capabil să învețe din experiența anterioară, adică din conversațiile și răspunsurile anterioare, și să își îmbunătățească astfel capacitatea de a genera răspunsuri precise și relevante.

Integrarea ChatGPT în aplicația web prin intermediul unei chei API a fost realizată prin integrarea cheii API în aplicația web - cheia API va fi utilizată pentru a face cereri către API-ul ChatGPT pentru a obține răspunsuri de la chatbot. Această integrare se poate realiza prin intermediul unui cod JavaScript care face apel la API-ul ChatGPT și trimite cererile cu cheia API inclusă în URL-ul de cerere.

Prin integrarea ChatGPT într-o altă aplicație web, utilizatorii aplicației vor putea interacționa cu chatbot-ul pentru a primi răspunsuri la întrebările lor sau pentru a rezolva probleme. Această integrare poate aduce o valoare adăugată semnificativă aplicației web și poate îmbunătăți experiența utilizatorilor prin oferirea unui serviciu de asistență în timp real și personalizat. În cadrul aplicației realizate, am configurat asistentul virtual să răspundă utilizatorului ca și cum ar fi un critic de film pentru o calitate mai bună și relevantă a răspunsurilor.

MongoDB Atlas este serviciul cloud de gestionare a bazelor de date MongoDB ce oferă o soluție scalabilă, performantă și sigură pentru stocarea datelor, care poate fi utilizată de dezvoltatori, echipa de operațiuni și de afaceri. Prin intermediul MongoDB Atlas, utilizatorii pot crea și gestiona baze de date MongoDB în cloud, fără a fi necesară instalarea și configurarea unui server propriu. Atlas oferă o interfață intuitivă pentru gestionarea bazei de date, oferind opțiuni de configurare avansată, cum ar fi backup și recuperarea datelor, replicare și redimensionare automată.

De asemenea, MongoDB Atlas oferă integrare cu diferite instrumente și servicii, cum ar fi Postman, pentru a facilita dezvoltarea și gestionarea aplicațiilor care utilizează baze de date MongoDB. MongoDB poate fi integrat cu Postman prin intermediul API-urilor REST oferite de baza de date. Acest lucru permite utilizatorilor să acceseze și să manipuleze datele din baza de date prin intermediul interfeței Postman.

Una dintre caracteristicile principale ale platformei Vercel este integrarea puternică cu platforma GitHub, care permite dezvoltatorilor să integreze cu ușurință aplicațiile lor cu sistemul de control al versiunilor GitHub. Acest lucru facilitează colaborarea între dezvoltatori și oferă un mediu de lucru sigur și ușor de gestionat.

În general, Vercel este o platformă puternică și ușor de utilizat pentru dezvoltarea și implementarea site-urilor web și a aplicațiilor web moderne. Prin intermediul instrumentelor sale de dezvoltare web, Vercel facilitează colaborarea și îmbunătățește productivitatea dezvoltatorilor, permițându-le să se concentreze pe dezvoltarea aplicațiilor de calitate superioară.

# Flux de date

Fluxul de date într-o aplicație web care utilizează două servicii în cloud prin intermediul unui API REST poate fi descris astfel:

* Clientul trimite o cerere HTTP la serverul API REST, specificând endpoint-ul și metoda HTTP dorite (GET, POST sau DELETE).
* Serverul API REST primește cererea HTTP și o procesează pentru a determina resursa solicitată și acțiunea corespunzătoare care trebuie efectuată asupra resursei.
* Serverul API REST utilizează apoi cele două servicii în cloud pentru a obține sau modifica datele necesare pentru a îndeplini cererea clientului.
* Datele sunt transferate între serviciile în cloud și serverul API REST prin intermediul protocolului HTTP și sunt transformate într-un format adecvat pentru a fi utilizate de către client.
* Serverul API REST compilează răspunsul și îl trimite înapoi la client prin intermediul HTTP. Răspunsul conține datele solicitate sau confirmarea acțiunii efectuate (de exemplu, ștergerea unei resurse).
* Clientul primește răspunsul HTTP și îl procesează, afișând datele sau mesajul de confirmare corespunzător.

În acest flux de date, API-ul REST este intermediarul între client și serviciile în cloud, permițând acestora să comunice într-un mod standardizat și eficient. Clientul nu trebuie să interacționeze direct cu serviciile în cloud, ceea ce reduce complexitatea și costurile dezvoltării și întreținerii aplicației web.

## Exemple de request / response

În MongoDB și Postman, elementele de request și response sunt similare cu cele ale altor baze de date și instrumente de testare a API-urilor.

Elemenete de request:

* Metoda HTTP: GET, POST, PUT, DELETE, PATCH etc.
* URI (Uniform Resource Identifier): identifică resursa dorită, cum ar fi o colecție sau un document în MongoDB.
* Parametrii de query: opționali și sunt folosiți pentru a specifica filtrarea, sortarea sau proiecția de date.
* Corpul mesajului: opțional și este utilizat în cazul în care se face o cerere care include date, cum ar fi inserarea sau actualizarea unui document în MongoDB.
* Antetele HTTP: conțin informații suplimentare despre cerere, cum ar fi tipul de conținut sau tokenul de autorizare.

Elemente de response:

* Codul de stare HTTP: indică dacă cererea a fost tratată cu succes sau dacă a avut loc o eroare, precum și tipul de răspuns (cum ar fi JSON sau XML).
* Antetele HTTP: conțin informații suplimentare despre răspuns, cum ar fi tipul de conținut sau lungimea răspunsului.
* Corpul mesajului: conține datele returnate de la MongoDB într-un format specificat (de obicei JSON sau BSON).

În Postman, aceste elemente pot fi introduse în modulul "Body" al unei cereri, iar răspunsul poate fi vizualizat în modulul "Response". În plus, Postman oferă și un istoric al cererilor și răspunsurilor pentru a urmări comunicarea cu baza de date MongoDB.

Am folosit trei constante specificate și în fișierul .env pentru a conecta aplicația noastră cu baza de date și cu OpenAI:

* NEXT\_ATLAS\_URI - unde am pus linkul către MongoDB si contul nostru
* NEXT\_ATLAS\_DATABASE - unde am atașat baza de date din Compass aplicației
* OPENAI\_API\_KEY - unde am conectat asistentul nostru virtual direct

## Metode HTTP

Metodele HTTP sunt verbele standard utilizate în comunicarea între client și server prin intermediul protocolului HTTP (Hypertext Transfer Protocol). Aceste metode descriu acțiunile pe care clientul dorește să le execute pe server și sunt incluse în cadrul unui request.

Metodele de HTTP utilizate în aplicație sunt:

* GET - solicită o resursă specifică de la server. Răspunsul este de obicei o reprezentare a resursei cerute. Nu trebuie să aibă un corp de mesaj.
* POST - trimite date către server pentru a fi procesate. Poate fi utilizat pentru a crea, actualiza sau șterge resurse. Corpul mesajului conține datele trimise către server.
* DELETE - șterge o resursă specifică de pe server. Nu trebuie să aibă un corp de mesaj.

Aceste metode sunt importante pentru dezvoltarea de API-uri, deoarece oferă o modalitate standardizată pentru comunicarea între client și server. De asemenea, permit manipularea resurselor și prelucrarea datelor într-un mod eficient și securizat.

Metoda HTTP GET este utilizată pentru a solicita o resursă specifică de la server. Cererea GET este de obicei utilizată pentru a obține informații și date de la server. Răspunsul este de obicei o reprezentare a resursei cerute. Metoda GET nu modifică starea serverului și nu trebuie să aibă un corp de mesaj.

Metoda HTTP POST este utilizată pentru a trimite date către server pentru a fi procesate. Cererea POST este adesea utilizată pentru a crea sau actualiza o resursă pe server. Corpul mesajului conține datele trimise către server. Metoda POST modifică starea serverului și poate să aibă un efect secundar, cum ar fi crearea unei resurse sau actualizarea unor date.

Metoda HTTP DELETE este utilizată pentru a șterge o resursă specifică de pe server. Cererea DELETE este adesea utilizată pentru a elimina o resursă pe care clientul nu o mai utilizează sau nu mai are nevoie de ea. Nu trebuie să aibă un corp de mesaj. Metoda DELETE modifică starea serverului și poate să aibă un efect secundar, cum ar fi ștergerea unei resurse sau actualizarea altor date.

# Capturi ecran aplicație

A screenshot of a movie screen

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, screenshot, font, number

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

# Referințe

* <https://cheatsheets.shecodes.io/javascript>
* <https://nextjs.org/docs>
* <https://legacy.reactjs.org/docs/getting-started.html>
* <https://www.youtube.com/watch?v=Tn6-PIqc4UM>
* <https://www.freecodecamp.org/news/the-beginners-guide-to-react-hooks/>
* <https://tailwindcss.com/docs/installation>
* <https://www.youtube.com/watch?v=mr15Xzb1Ook>
* <https://www.mongodb.com/docs/drivers/node/current/fundamentals/crud/>
* <https://platform.openai.com/docs/introduction>
* <https://learning.postman.com/docs/getting-started/sending-the-first-request/>
* <https://www.codecademy.com/learn/learn-node-js/modules/intro-to-node-js/cheatsheet>