

Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Aplicație WEB pentru căutare de hoteluri – Python

Student
Dobre Alexandru-Cristian
Grupa
443A

Profesor coordonator
Conf.dr.ing. Bogdan Florea

București, 2024

Cuprins

I.	Introducere	3
II.	Tehnologii folosite	4
III.	Prototipul aplicației	5
IV.	Implementare	8
V.	Concluzie	11
VI.	Bibliografie	12

I. Introducere

Despre Aplicația de Căutare a Hotelurilor

Aplicația de Căutare a Hotelurilor este o soluție web inovatoare care își propune să simplifice procesul de căutare și rezervare a hotelurilor pentru călătorii. Dezvoltată cu ajutorul tehnologiei Flask, un framework web ușor și flexibil pentru Python, această aplicație oferă utilizatorilor o interfață intuitivă pentru a găsi cazare în diverse locații, bazată pe criterii precum locația, datele de check-in și check-out, și numărul de persoane adulte.

Funcționalități Principale

- 1)Căutarea Hotelurilor: Utilizatorii pot introduce un oraș și pot specifica intervalul de date pentru cazare și numărul de adulți pentru a căuta hoteluri disponibile.
- 2)Vizualizarea Ofertelor de Camere: Aplicația preia și afișează ofertele detaliate ale camerelor disponibile în hotelurile selectate, inclusiv numele camerei, prețul și poza cu aceasta.
- 3)Extragerea Scorurilor de Recenzie și Ratingurilor: Pe lângă detaliile camerelor, aplicația aduce și scoruri de recenzie și ratinguri pentru fiecare hotel, oferind astfel utilizatorilor informații suplimentare pentru a lua decizii informate.
- 4)Sortarea Rezultatelor: Utilizatorii pot sorta rezultatele căutării în funcție de preț sau rating, atât în ordine crescătoare, cât și descrescătoare, facilitând astfel găsirea celei mai potrivite opțiuni de cazare.

II. Tehnologii utilizate

Flask: Un micro-framework pentru Python, folosit pentru dezvoltarea backend-ului aplicației.

HTML/CSS: Utilizate pentru structurarea și stilizarea paginii web.

JavaScript: Folosit pentru a adăuga interactivitate pe partea de client, inclusiv pentru sortarea rezultatelor căutării în funcție de diferite criterii.

API-uri Externe: Aplicația se integrează cu un API extern pentru a obține date despre hoteluri, inclusiv informații despre camere, prețuri și recenzii.

III. Prototipul Aplicatiei

Pagina Principala – Formular de Cautare

Căutare Hoteluri

Oraș:

Bucharest

Data Check-in:

02 / 20 / 2024



Data Check-out:

02 / 23 / 2024



Adulți:

2

Caută

Cautare in datele returnate de endpoint-ul `locations/v3/search` pe baza orasului

```
Hotels Data Received
{'q': 'Bucharest', 'rid': '165d193e3eea4cfd8dfd261e44420438', 'rc': 'OK', 'sr': [{'@type': 'gaiaRegionResult', 'index': '0', 'gaiaId': '6057278', 'type': 'CITY',
  'regionNames': {'fullName': 'Bucharest, Romania', 'shortName': 'Bucharest', 'displayName': 'Bucharest, Romania', 'primaryDisplayName': 'Bucharest', 'secondaryDisplayName':
  'Romania', 'lastSearchName': 'Bucharest'}, 'essId': {'sourceName': 'GAI', 'sourceId': '6057278'}, 'coordinates': {'lat': '44.426769', 'long': '26.102539'}, 'hierarchyInfo':
  {'country': {'name': 'Romania', 'isoCode2': 'RO', 'isoCode3': 'ROU'}, 'airport': {'airportCode': 'BUH', 'airportId': '6139079', 'metrocode': 'BUH', 'multicity': '719'}}},
  {'@type': 'gaiaRegionResult', 'index': '1', 'gaiaId': '553248633981727536', 'type': 'NEIGHBORHOOD', 'regionNames': {'fullName': 'Bucharest City-Centre, Bucharest, Romania',
  'shortName': 'Bucharest City-Centre', 'displayName': 'Bucharest City-Centre, Bucharest, Romania', 'primaryDisplayName': 'Bucharest City-Centre', 'secondaryDisplayName':
  'Bucharest, Romania', 'lastSearchName': 'Bucharest City-Centre'}, 'essId': {'sourceName': 'GAI', 'sourceId': '553248633981727536'}, 'coordinates': {'lat': '44
  .43974490326842', 'long': '26.09905775621643'}, 'hierarchyInfo': {'country': {'name': 'Romania', 'isoCode2': 'RO', 'isoCode3': 'ROU'}, 'airport': {'airportCode': 'BUH',
  'airportId': '6139079', 'metrocode': 'BUH', 'multicity': '719'}}}, {'@type': 'gaiaRegionResult', 'index': '2', 'gaiaId': '6346485', 'type': 'NEIGHBORHOOD', 'regionNames':
  {'fullName': 'Bucharest Old Town, Bucharest, Romania', 'shortName': 'Bucharest Old Town', 'displayName': 'Bucharest Old Town, Bucharest, Romania', 'primaryDisplayName':
  'Bucharest Old Town', 'secondaryDisplayName': 'Bucharest, Romania', 'lastSearchName': 'Bucharest Old Town'}, 'essId': {'sourceName': 'GAI', 'sourceId': '6346485'},
  'coordinates': {'lat': '44.432330763414136', 'long': '26.102956302911235'}, 'hierarchyInfo': {'country': {'name': 'Romania', 'isoCode2': 'RO', 'isoCode3': 'ROU'}, 'airport':
  {'airportCode': 'BUH', 'airportId': '6139079', 'metrocode': 'BUH', 'multicity': '719'}}}, {'@type': 'gaiaRegionResult', 'index': '3', 'gaiaId': '5264463', 'type': 'AIRPORT',
  'regionNames': {'fullName': 'Bucharest, Romania (OTP-Henri Coanda Intl.)', 'shortName': 'Bucharest (OTP-Henri Coanda Intl.)', 'displayName': 'Bucharest (OTP - Henri Coanda
  Intl., Romania', 'primaryDisplayName': 'Bucharest (OTP - Henri Coanda Intl.)', 'secondaryDisplayName': 'Romania', 'lastSearchName': 'Bucharest (OTP - Henri Coanda Intl.)'},
  'essId': {'sourceName': 'GAI', 'sourceId': '5264463'}, 'coordinates': {'lat': '44.57003', 'long': '26.07578'}, 'hierarchyInfo': {'country': {'name': 'Romania', 'isoCode2':
  'RO', 'isoCode3': 'ROU'}, 'airport': {'airportCode': 'OTP', 'airportId': '5264463', 'metrocode': 'BUH', 'multicity': '719'}, 'isMinorAirport': 'false'}, {'@type':
  'gaiaHotelResult', 'index': '4', 'hotelId': '34928289', 'type': 'HOTEL', 'regionNames': {'fullName': 'Hilton Garden Inn Bucharest Airport, Otopeni, Ilfov, Romania',
  'shortName': 'Hilton Garden Inn Bucharest Airport', 'displayName': 'Hilton Garden Inn Bucharest Airport, Otopeni, Ilfov, Romania', 'primaryDisplayName': 'Hilton Garden Inn
```

Se obtin `RegionId`, `HotelId`, coordonate care sunt trimise in apelul metodei `get_hotel_offers` sau `get_reviews` care obtin date in format JSON de la endpoint-urile `properties/v2/get-offers`", respectiv `reviews/v3/get-summary`"

Se cauta in `get-offers` ofertele disponibile pe baza campurilor completate

```
Hilton Garden Inn Bucharest Airport, Otopeni, Ilfov, Romania
hotel id: 34928289
region id: 603282
Checking for OFFERS...
ADULTS: 2
ROOM NAME: Room, 2 Twin Beds
PRICE: $380 for 3 nights
ROOM NAME: Room, 1 King Bed
PRICE: $380 for 3 nights
ROOM NAME: Room, 1 King Bed, Accessible
PRICE: $380 for 3 nights
ROOM NAME: Double Room King Bed
PRICE: $460 for 3 nights
ROOM NAME: Twin Room
PRICE: $460 for 3 nights
ROOM NAME: Double Room King Bed - Accessible
PRICE: $460 for 3 nights
ROOM NAME: Family Room, 1 King Bed
PRICE: $502 for 3 nights
ROOM NAME: Room, 1 King Bed with Sofa bed
PRICE: $502 for 3 nights
```

```
JW Marriott Bucharest Grand Hotel, Bucharest, Romania
hotel id: 570476
region id: 6057278
Checking for OFFERS...
ADULTS: 2
ROOM NAME: Deluxe Room, 1 King Bed, Non Smoking
PRICE: $819 for 3 nights
ROOM NAME: Deluxe Room, 2 Double Beds, Non Smoking
PRICE: $819 for 3 nights
ROOM NAME: Superior Room, 2 Double Beds, Non Smoking
PRICE: $759 for 3 nights
ROOM NAME: Superior Room, 1 King Bed with Sofa bed, Non Smoking
PRICE: $759 for 3 nights
ROOM NAME: Executive Room, 1 King Bed, Non Smoking
PRICE: $885 for 3 nights
ROOM NAME: Executive Room, 2 Double Beds, Non Smoking
PRICE: $753 for 3 nights
ROOM NAME: Grand Room, 1 King Bed with Sofa bed, Non Smoking
PRICE: $959 for 3 nights
```

Datele preluate sunt apoi prelucrate si afisate intr-o pagina web cu optiunea de sortare dupa:

- 1) Pret
- 2) Rating

Exemplu de afisare:

Rezultate Cautare Hoteluri

[Sortează Pret Crescător](#)[Sortează Pret Descrescător](#)[Sortează Rating Crescător](#)[Sortează Rating Descrescător](#)

Hotel: Prague Marriott Hotel, Prague, Czechia

Camera: Deluxe Room, 2 Queen Beds, Non Smoking

Pret: \$1,504 for 4 nights USD

Rating: 92

[Vezi Poza](#)

Hotel: Prague Marriott Hotel, Prague, Czechia

Camera: Deluxe Room, 1 Queen Bed, Non Smoking

Pret: \$1,504 for 4 nights USD

Rating: 92

[Vezi Poza](#)

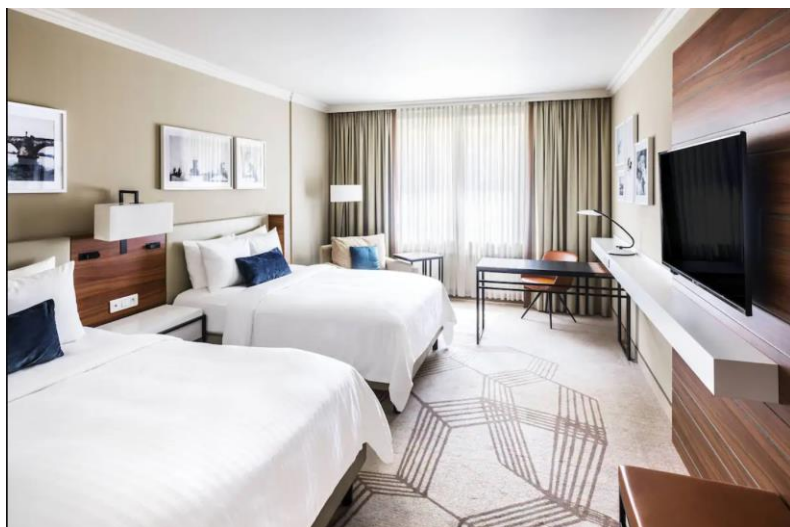
Hotel: Prague Marriott Hotel, Prague, Czechia

Camera: Superior Room, 2 Queen Beds, Non Smoking

Pret: \$1,802 for 4 nights USD

Rating: 92

Apasand pe butonul “Vezi poza”, utilizatorul va putea vedea poze cu camera respectiva



IV. Implementare

Backend

1) Configurarea Aplicației Flask

```
app = Flask(__name__)
```

Inițializează o instanță a aplicației Flask.

2) Configurarea Anteturilor pentru API

Setează anteturile necesare pentru solicitările către API-ul de hoteluri

3) Ruta Principală - Afișarea Formularului de Căutare

```
@app.route('/')
def index():
    return render_template('index.html')
```

Definește ruta principală ("/") care servește pagina de pornire.
index.html ar trebui să conțină formularul de căutare pentru hoteluri.

4) Ruta de Căutare - Procesarea Solicitării Formularului

```
@app.route('/search', methods=['POST'])
def search():
    # extrage datele din formular
    # efectuează solicitarea la API
    # procesează răspunsul
    # returnează 'results.html' cu datele prelucrate
```

Această rută gestionează solicitările de căutare trimise de utilizatori.
Extragerea datelor din formular și efectuarea unei solicitări la API pentru a găsi hoteluri în orașul specificat.

5) Solicitarea la API-ul de Căutare Hoteluri

```
response = requests.get(url, headers=headers, params=querystring)
```


Efectuează o solicitare GET la API pentru a obține informații despre hoteluri bazate pe querystring (orașul selectat și setările locale).

6)Procesarea Răspunsului API și Extragerea Ofertelor

```
for item in hotels_data.get('sr', []):  
    # procesează fiecare hotel găsit  
    # obține ofertele fiecărui hotel folosind `get_hotel_offers`
```

Iterează prin rezultatele căutării pentru a obține detalii despre fiecare hotel, cum ar fi ID-ul și coordonatele.

Apelurile către `get_hotel_offers` și `get_hotel_review_score` obțin ofertele și scorurile de recenzie pentru fiecare hotel.

7)Metoda `get_hotel_offers`

```
def get_hotel_offers(checkin, checkout, adults, hotel_id, latitude, longitude, region_id):  
    # efectuează o solicitare la API pentru a obține ofertele hotelului
```

Trimite o solicitare POST la API pentru a obține ofertele specifice ale unui hotel, inclusiv detalii despre camere și prețuri.

8)Extragerea Numele Camerei, Prețului si Pozei

```
for unit in offers_data['data']['propertyOffers']['units']:  
    with open('output3.json', 'w') as file:  
        json.dump(unit, file, indent=4)  
        room_name = unit.get('header', {}).get('text') # Numele camerei  
        print("ROOM NAME: ", room_name)  
        for rate_plan in unit.get('ratePlans'):  
            for room_price in rate_plan['priceDetails']:  
                price = room_price['totalPriceMessage']  
                break  
            print("PRICE: ", price)  
            for pic_plan in unit.get('unitGallery', {}).get('gallery'):  
                pic = pic_plan.get('image', {}).get('url')  
                break  
            # print(f'pic url: {pic}')  
            room_offers.append({'room_name': room_name, 'price': price, 'pic': pic})
```

9)Metoda get_hotel_review_score

```
def get_hotel_review_score(hotel_id):
```

efectuează o solicitare la API pentru a obține scorul de recenzie al hotelului
Solicită scorul de recenzie și procentajul de rating pentru un hotel specific folosind ID-ul acestuia. Această informație este folosită pentru a oferi utilizatorilor o imagine de ansamblu a calității hotelului.

10)Rularea Aplicației Flask

```
if __name__ == '__main__':  
    app.run(debug=True)
```

Frontend

1)index.html - Formularul de Căutare

<!DOCTYPE html>: Specifică tipul documentului și versiunea HTML.

<html>: Tagul rădăcină pentru orice document HTML.

<head>: Conține metadatele documentului, cum ar fi titlul.

<title>Căutare Hoteluri</title>: Titlul paginii, afișat în bara de titlu a browserului.

<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">: Include un fișier CSS pentru stilizarea paginii.

<h1>Căutare Hoteluri</h1>: Un titlu principal pentru pagina de căutare.

<form action="/search" method="post">: Definiște un formular cu metoda POST către ruta /search.

Elemente input pentru introducerea datelor de căutare (oraș, datele de check-in/check-out, numărul de adulți).

Butonul submit trimite datele formularului.

2)results.html - Afișarea Rezultatelor Căutării

<script>: Conține cod JavaScript pentru a sorta hotelurile. Funcția sortHotels sortează elementele în listă în funcție de preț sau rating, în ordine crescătoare sau descrescătoare.

Butonul onclick: Fiecare buton are un atribut onclick care apelează sortHotels cu parametrii corespunzători (tipul de sortare și ordinea).

<ul id="hotelsList">: O listă neordonată unde fiecare hotel este reprezentat printr-un element li.

{% for hotel in hotels %}: O buclă care iterează prin lista de hoteluri trimisă de Flask.

data-price și data-rating: Atribute personalizate pentru stocarea prețului și ratingului, folosite în sortare.

Afisează numele hotelului, numele camerei, prețul, ratingul și un link către imagine.

V. Concluzie

Această aplicație Flask demonstrează cum se pot combina informațiile de la diverse API-uri pentru a crea o experiență utilă și informativă pentru utilizatori. Fluxul de date de la cererea inițială până la prezentarea finală în browser este bine structurat, oferind o soluție integrată pentru căutarea și vizualizarea opțiunilor de cazare.

Integrarea Eficientă a API-urilor

Aplicația utilizează eficient datele furnizate de API-uri externe pentru a oferi informații actualizate și relevante despre hoteluri. Prin combinarea mai multor surse de date (informații despre hoteluri, recenzii, prețuri, imagini), aplicația oferă o perspectivă cuprinzătoare asupra opțiunilor de cazare disponibile, permițând utilizatorilor să facă alegeri informate bazate pe o varietate de factori, inclusiv preț, locație și calitatea generală a hotelului.

Ușurința de Utilizare

Interfața simplă și intuitivă a aplicației permite utilizatorilor de toate nivelurile de experiență să navigheze și să utilizeze eficient funcțiile disponibile. De la formularul de căutare inițial până la afișarea rezultatelor sortate, fluxul aplicației este logic și ușor de urmărit.

Flexibilitate și Personalizare

Funcționalitățile de sortare adăugate, cum ar fi sortarea după preț și rating, oferă utilizatorilor controlul asupra modului în care doresc să vadă și să compare rezultatele. Această personalizare îmbunătățește experiența utilizatorului, permițând fiecăruia să găsească ceea ce caută într-un mod mai eficient și personalizat. Posibilitatea de a vedea imagini direct din rezultatele căutării adaugă un nivel suplimentar de interactivitate și implicare pentru utilizatori, făcând procesul de căutare nu doar informativ, dar și vizual atrăgător.

Fiabilitate și Actualizare Continuă

Datorită integrării cu API-uri externe, aplicația poate furniza informații actualizate și de încredere. Aceasta este esențială în industria hotelieră, unde disponibilitatea și prețurile se pot schimba rapid. Utilizatorii beneficiază de accesul la cele mai recente date disponibile, asigurându-se că deciziile lor sunt bazate pe cele mai actuale și precise informații.

Extensibilitate și Scalabilitate

Structura modulară a aplicației și utilizarea Flask ca framework oferă un temel solid pentru extindere și scalabilitate. Pe măsură ce nevoile utilizatorilor evoluează sau dacă doriți să adăugați noi funcționalități, aplicația poate fi adaptată și extinsă relativ ușor. Aceasta include posibilitatea de a adăuga noi surse de date, funcționalități suplimentare de filtrare sau chiar integrarea cu alte servicii, cum ar fi rezervarea directă a camerei de hotel.

VI. Bibliografie

<https://rapidapi.com/apidojo/api/hotels4> - informatii despre endpoint-uri
<https://www.dataquest.io/blog/python-api-tutorial/>
https://www.w3schools.com/js/js_json_intro.asp