Capital One Hackathon Project

Made by: *Los Pepinos*

Prompt Idea

The travel planning app industry has seen substantial growth in recent years, fueled by rising demand for personalized and convenient travel experiences. In this context, Capital One aims to enhance the travel experience for its credit cardholders by developing an app that utilizes transaction data and othe parameters to generate personalized travel itineraries.

Introduction

We started researching information from the current market to use them as guidelines to what ideas could be implemented.

- Pricing ranges
- Destinations
- Flight deals
- And more...

We also studied our own capabilities and deficiencies to get a better understanding on where we stand as a company.

- Better customer information
- Capital One Travel

After our research we noticed we could focus on two possible paths to implement the itinerary recommendation.

- Create and implement a subsection inside the already created Capital One Travel app/website.
- Create a new app/website with the implementation.

Pricing

- Regarding implementation costs, we cannot provide a specific figure due to the variability that could arise depending on whether we start from scratch or not. (*The two paths*)
- Capital One does not disclose the investments made in the development of its applications or website, and for these reasons, we cannot provide a specific figure.
- Utilizing Capital One Travel's existing infrastructure has both pros and cons compared to starting from scratch. These aspects are crucial to evaluate the path forward in our product proposal.

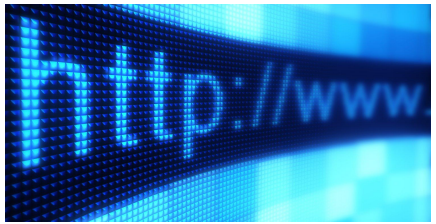
Pros of implementing on Capital One Travel platforms

Implementing the itinerary recommendations as an added section on Capital One Travel could:

- **Lower Costs of Initial Development–** Developing a subcategory within a webpage generally requires less time and resources than creating an entirely new application.
- **Lower Costs of maintenance–** Once the subcategory is integrated in the webpage, ongoing maintenance is more economical than maintaining an independent application.
- **Easier future implementations or updates** – You can implement changes and new features more quickly and efficiently

Pros of Implementing on New App/Website

- It would allow for clearer analytical data since having a dedicated website makes easier to track and analyze metrics.
- **Expandability and Flexibility:** Building a new web page offers greater flexibility in terms of feature expansion and future updates.
- **Branding and Marketing:** A new web page provides an opportunity for distinct branding and marketing efforts.
- **Technical Considerations:** Separating the travel itinerary recommendation service into its own web page can simplify technical implementation and maintenance.



Competition

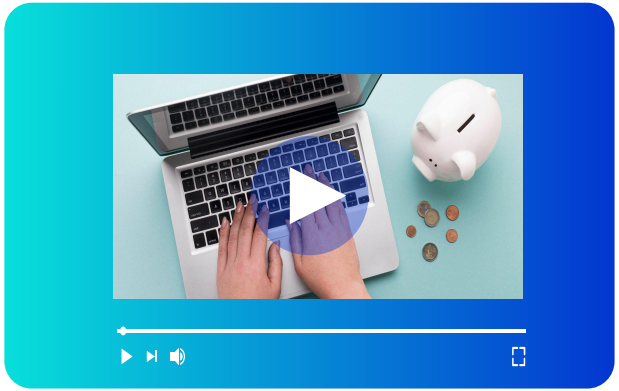
- Capital One competes in the travel planning app market against industries such as TripAdvisor, Expedia, Airbnb, TripIt and Hopper.
- In the realm of personalized travel itinerary apps, Capital One faces competition from several companies offering similar solutions:
    - **Hopper**: Utilizes AI to predict flight prices and recommend optimal booking times. Offers personalized itinerary creation based on user preferences.
    - **TripIt**: Allows users to create personalized travel itineraries from their booking confirmation emails. Users can also share itineraries with friends and family.
    - **Kayak**: A travel platform enabling users to search and book flights, hotels, and rental cars. Provides options for creating personalized itineraries and receiving price alerts.
    - **Expedia**: A travel platform allowing users to search and book flights, hotels, and rental cars. Offers personalized itinerary creation and price alerts.

Capital One, it's different!

- Differentiation will be key to Capital One's success in this space, emphasizing the app's ability to generate personalized itineraries based on financial and transaction data.
- **Advantages of Capital One:**
  - Detailed financial data access: Capital One utilizes detailed transaction data from its credit cardholders to understand spending patterns, enabling highly personalized travel recommendations.
  - Financial product integration: Capital One's travel app integrates seamlessly with its other financial products, like travel credit cards and rewards programs, bookings and offering exclusive benefits.
  - Consumer security focus: Capital One prioritizes consumer security and protection throughout the travel planning process, including real-time security notifications, fraud protection, and data privacy guarantees.

Possible Weaknesses of Our Own App/Website:

- **Privacy and Security Concerns:** Mishandling sensitive financial data access could raise privacy and security issues.
- **Poor User Experience:** A non-intuitive interface or usability issues may drive users towards simpler, competitor-provided solutions.
- **Technological Limitations:** Lack of robust technological infrastructure or key features like accurate and timely recommendations may frustrate users, leading them to explore alternative options.



Prototype Development

***The Prototype we implemented is based on 3 subphases*:**

Subphase 1: Data Management

- Use the wealth of data stored in Capital One's pre-existing databases as the foundation for our prototype.
- Explore transactional records, spending patterns, and user behaviors to inform the personalized travel planning features.
- Consider a more advanced future implementation by incorporating NESSIE, Capital One's simulated database, for cutting-edge testing and refinements.
- NESSIE provides a simulated environment to test the prototype with real scenarios, ensuring robustness and adaptability.

How we managed the data


```
{
      "Reservations": [
          {
              "OwnerId": "",
              "ReservationId": "",
              "Groups": [],
              "Instances": [
                  {
                      "Monitoring": {
                          "State": "disabled"
                      },
                      "PublicDnsName": "",
                      "State": {
                          "Code": 16,
                          "Name": "running"
                      },
                      "EbsOptimized": false,
                      "LaunchTime": "2016-08-31T22:39:37.000Z",
                      "PublicIpAddress": "",
                      "PrivateIpAddress": "",
                      "ProductCodes": [],
                      "VpcId": "",
                      "StateTransitionReason": "",
                      "InstanceId": "",
                      "ImageId": "",
                      "PrivateDnsName": "",
                      "KeyName": "",
                      "SecurityGroups": [...
```

Managing the Data

```
#Imitation of a JSON Data transformed into a dictionary
UserData = {
  "User": "John Doe",
  "Age": 30,
  "City": "New York",
  "Anual Debt": 5000,
  "Anual Income": 80000,
  "Average expenses of Needs": 10000,
  "Average Savings": 8000,
  "Loans": 3000,
  "Credit": 6000,
  "Investments": 7000,
  "Balance": 40000,
  "Montly Expenses": 0,
  "Monthly Income": 10000,
  "Monthly Savings": 8000,
  "Monthly Loans": 3000
}
#####Extract User Data##########################################

res=UserPercentageFiltration(UserData)
```

Filtering the data

```
39  def UserPercentageFiltration(data):
40    result = {"Travel Budget": 0, "Left Money": 0, "Used week": "", "Vacation Itinerary Percentage": 0}
41    for k, v in data.items():
42        if k == "Anual Income":
43            result["Travel Budget"] = v
44
45            if v >= 50000 and v<= 100000:
46              vacation_percentage = 0.15
47              result["Travel Budget"] = v * vacation_percentage
48
49            elif v >= 25000 and v< 50000:
50              vacation_percentage = 0.10
51              result["Travel Budget"] = v * vacation_percentage
52
53            elif v < 25000:
54              vacation_percentage = 0.5
55              result["Travel Budget"] = v * vacation_percentage
56
57            result["Vacation Itinerary Percentage"] = vacation_percentage
58    result["Left Money"] = data["Anual Income"] - result["Travel Budget"]
59    result["Vacation Itinerary Percentage"]= vacation_percentage *100
60    result["City"] = data["City"]
61    return result
62
63  def TripDays(option, filteredData):
64    if(option=="Weekends"):
65      filteredData["Used week"]= "Weekends"
66    elif(option=="Weekdays"):
67      filteredData["Used week"]= "Weekdays"
68    elif (option=="Seasonal"):
69      filteredData["Used Week"]="Seasonal"
70    return filteredData
71
72
```

Prototype Development

Subphase 2: Search Engine

- Since our code has the capacity to search for information on the internet it does retreat and helps with the filtering for itinerary purposes.
- According to some limited resources Capital One currently lacks of direct affiliations with travel agencies or airlines and our prototype creates a promising ground for exploring strategic partnerships.

Implementation of the Search Engine

```python
from googleapiclient.discovery import build
import time
import requests
from bs4 import BeautifulSoup
import re



def google_search(api_key, search_engine_id, query):
    service = build("customsearch", "v1", developerKey=api_key)
    res = service.cse().list(q=query, cx=search_engine_id).execute()
    return res['items']



def scrape_expedia_deal(url):


  # Extracting hotel name
  # hotel_name = soup.find('h1', class_='uitk-type-heading-500').text.strip()
  hotel_name="Dominican Fiesta Hotel"

  budget=392
  stay_duration="Weekend"

  things_to_do=["Pool","Restaurants","Bars"]

  # Building dictionary with scraped information
  expedia_deal_info = {
      'Destination': url,
      'hotel': hotel_name,
      'budget': budget,
      'stay_duration': stay_duration,
      'things_to_do': things_to_do[:3]  # Limiting to the first 3 things to do
  }
  return expedia_deal_info
```

Search engine way to access

```python
# Info that lets us search in google, searching terms and searched data
api_key = 'AIzaSyB9Pb1pVGt5n1o0i6DIqMWqcaQl80uEF5s'
search_engine_id = '360a3807299d24b9e'
query_terms = ['All_inclusive','Expedia','Weekend']
results = google_search(api_key, search_engine_id, query_terms)

# Procesar los resultados de la búsqueda
for result in results:
    title = result['title']
    url = result['link']
    print(title,url)
    print()
    time.sleep(2)

results=TripDays("Weekends",res)
print()
print()
Itinerary_Trip(results,392,"Dominican Fiesta Hotel")
print()
x=input("Do you want to save this trip? ")
print()
if(x=="yes"):
  try:
    deal_info = scrape_expedia_deal(url)
    print(deal_info)
  except Exception as e:
    print(f"Error scraping {url}: {str(e)}")
  print()
  print("This info has been sent to mail")
  print("Thank you for using our service")
else:
  print("Thank you for using our service")
```

Prototype Development

Subphase 3: Itinerary Creation

- Based on the acquired data and the filtered user information, we developed a product that let us provide an itinerary based on the customers preferences and algorithms.

```
results=TripDays("Weekends",res)
print()
print()
Itinerary_Trip(results,392,"Dominican Fiesta Hotel")
print()
x=input("Do you want to save this trip? ")
print()
if(x=="yes"):
  try:
    deal_info = scrape_expedia_deal(url)
    print(deal_info)
  except Exception as e:
    print(f"Error scraping {url}: {str(e)}")
  print()
  print("This info has been sent to mail")
  print("Thank you for using our service")
else:
  print("Thank you for using our service")
```

```
Itinerary:
New York to Dominican Fiesta Hotel
Stay: Weekends
Travel Cost: 392
```

Our Learning Experience!

Thanks!

Feel free to ask us any questions!!!