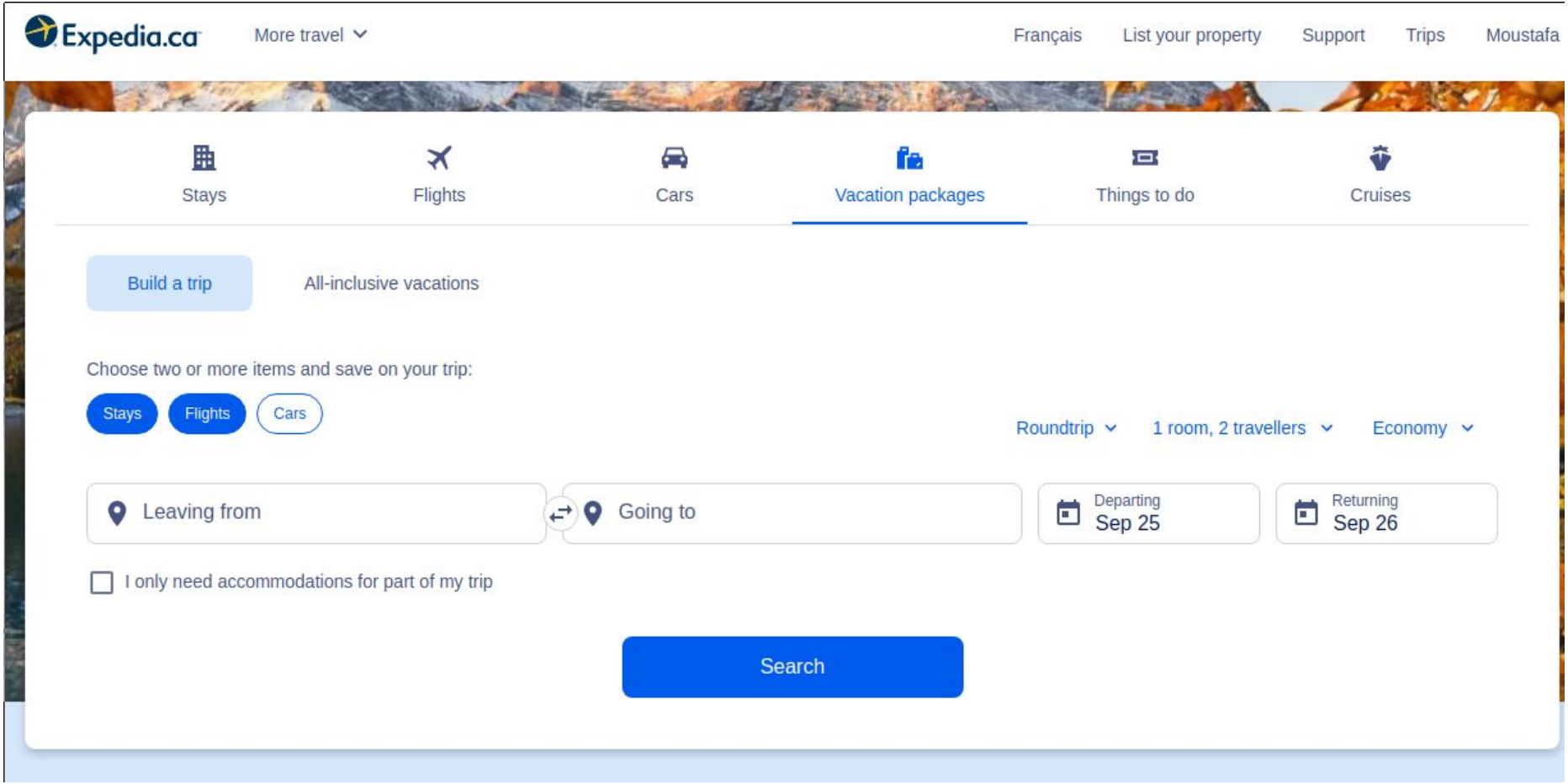
Python Programming

Expedia.com Project

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)


Expedia.ca
More travel
Français
List your property
Support
Trips
Moustafa
Stays
Flights
Cars
Vacation packages
Things to do
Cruises
Build a trip
All-inclusive vacations
Choose two or more items and save on your trip:
Stays
Flights
Cars
Roundtrip
1 room, 2 travellers
Economy
Leaving from
Going to
Departing
Sep 25
Returning
Sep 26
I only need accommodations for part of my trip
Search

Functional Requirements

- In expedia, a user creates several itineraries, each **itinerary** consists of several **reservations** as following
  - 0 or more flights, hotels, cars, etc. E.g. 4 flights, 2 hotels and 2 cars.
- Each reservation may has its own info
  - E.g. Hotel cost: total nights x price per night
- The itinerary cost = sum of its inner reservations
- For simplicity
  - Don’t use files / Use dummy data when convenient + store in memory
  - 2 types of users: Admin & customer. Focus of your code is on Customer part
  - Don’t burn time validating inputs or minor concerns
  - The core goal is OOP design skills

APIs

- Expedia needs to contact several APIs
- Flights APIs
  - Companies such as AirCanada, TurkishAirlines and others allow them to do online query to get current available flights
  - Then after the customer make a choice, you ask them to cancel/reserve
- Hotel APIs: In a similar way, hotels such as Hilton, Marriott provide APIs
- Payments: Expedia uses one of the payments APIs (e.g. Square/Stripe, etc)
- Follow the homework. Your code should be **loosely coupled** with these APIs
- Your code should be extensible: Future similar APIs might be used
- Content of the APIs is not hours. Put **dummy data to simulate**
- APIs code is given. Download it.

Login interface

```
System Access:
1) Login
2) Sign up [NA]
Enter your choice (from 1 to 2): 1
Enter username: most
Enter password: mo11

Welcome Mostafa Saad | Customer:
1) View profile [NA]
2) Make itinerary
3) List my itineraries
4) Logout
Enter your choice (from 1 to 4): 2
```

- I added a dummy user to use for login
  - Do proper validation for the login part
  - Initially, skip sign up
- The user can do 4 major actions
  - Step 2 and 3 are major goals
- Let’s make an itinerary (choice 2)

Make itinerary Interface

```
Create your itinerary:
1) Add Flight
2) Add Hotel
3) Reserve itinerary
4) Cancel itinerary
```

- The user can add 0 or more flights. Same for hotels
  - This menu will keep appearing to add as much as the user wants
- The user can reserve all added items or cancel all
  - Once user is done, he either choose 3 or 4
- Let’s add an hotel (choice 2)

Add hotel

- Enter hotel info. Search through hotel APIs and list them, then user choose
- In a similar way, we can do the add flight

```
Enter your choice (from 1 to 4): 2
Enter room type: grandview
Enter From Date (dd-mm-yy): 20-10-21
Enter To Date (dd-mm-yy): 15-12-21
Enter location: vancouver
Enter # of rooms: 1
Enter # of children: 2
Enter # of adults: 2

Select a hotel:
1) Hilton: Per night: 200.0 - Total cost 1000.0 - From: 1 on 20-10-21 - #num nights 5 - #num rooms 2 - #children 2 #adults 2
2) Hilton: Per night: 300.0 - Total cost 1500.0 - From: 1 on 20-10-21 - #num nights 5 - #num rooms 2 - #children 2 #adults 2
3) Hilton: Per night: 500.0 - Total cost 2500.0 - From: 1 on 20-10-21 - #num nights 5 - #num rooms 2 - #children 2 #adults 2
4) Marriott: Per night: 444.0 - Total cost 2220.0 - From: 1 on 20-10-21 - #num nights 5 - #num rooms 2 - #children 2 #adults 2
5) Marriott: Per night: 350.0 - Total cost 1750.0 - From: 1 on 20-10-21 - #num nights 5 - #num rooms 2 - #children 2 #adults 2
Enter your choice (from 1 to 5): 5
```

Add flight

```
Create your itinerary:
1) Add Flight
2) Add Hotel
3) Reserve itinerary
4) Cancel itinerary
Enter your choice (from 1 to 4): 1
Enter from: cairo
Enter From Date (dd-mm-yy): 20-10-21
Enter To: vancouver
Enter Return Date (dd-mm-yy): 15-12-21
Enter # of infants: 1
Enter # of children: 2
Enter # of adults: 2

Select a flight:
1) AirCanada: Cost 200 - From: cairo on 20-10-21 - To vancouver on 15-12-21 - #infants 1 #children 2 # adults 2
2) AirCanada: Cost 250 - From: cairo on 20-10-21 - To vancouver on 15-12-21 - #infants 1 #children 2 # adults 2
3) Turkish Airline: Cost 400 - From: cairo on 20-10-21 - To vancouver on 15-12-21 - #infants 1 #children 2 # adults 2
4) Turkish Airline: Cost 431 - From: cairo on 20-10-21 - To vancouver on 15-12-21 - #infants 1 #children 2 # adults 2
Enter your choice (from 1 to 4): 3
```

Reserve and Pay

- If the user decided to reserve the itinerary, then he should pay
- The customer profile has added payment cards. Let him choose

```
Create your itinerary:
1) Add Flight
2) Add Hotel
3) Reserve itinerary
4) Cancel itinerary
Enter your choice (from 1 to 4): 3

Which payment card:
1) DebitCard- Name: Mostafa, Number: 1234, Expiry Date: 09-2035
2) CreditCard- Name: Mostafa, Number: 4321, Expiry Date: 09-2035
Enter your choice (from 1 to 2): 2
PayPalOnlinePaymentAPI pay_money
Successfully paid and reserved the trip
```

More on payment logic

- Let’s say the user itinerary is 2 flights and 3 hotels reservation
- First, sum the total cost of all of them. Say $5000
- Contact the payment API to pay money
  - It may fail if no enough money or any network/system error
  - If it passed, start to reserve the 5 reservations
    - Say after reserving 2 flights and 1 hotel, then next reservation failed!
      - You need to cancel the payment
      - You need to cancel the 3 reservations
      - For simplicity: assume cancellation always works with no issues
- You may raise reasonable exceptions for different errors

Listing your itineraries

- So far we have 1. Let’s print it

```
Welcome Mostafa Saad | Customer:
1) View profile [NA]
2) Make itinerary
3) List my itineraries
4) Logout
Enter your choice (from 1 to 4): 3
Listing 1 itineraries
Itinerary Total Cost 2150.0
	Marriott: Per night: 350.0 - Total cost 1750.0 - From: 1 on 20-10-21 - #num nights 5 - #num rooms 2 - #children 2 #adults 2
	Turkish Airline: Cost 400 - From: cairo on 20-10-21 - To vancouver on 15-12-21 - #infants 1 #children 2 # adults 2
```

Flights APIs: AirCanada

```python
class AirCanadaCustomerInfo:...

class AirCanadaFlight:
    def __init__(self, price, date_time_from, date_time_to):...

class AirCanadaOnlineAPI:
    @staticmethod
    def get_flights(from_loc, from_date, to_loc, to_date, adults, children):
        flights = []
        flights.append(AirCanadaFlight(200, "25-01-2022", "10-02-2022"))
        flights.append(AirCanadaFlight(250, "29-01-2022", "10-02-2022"))
        return flights

    @staticmethod
    def reserve_flight(flight: AirCanadaFlight, customers_info: list):
        confirmation_id = '1234AirCanadaXXr34'  # None for failure
        return confirmation_id
        #return None     # Try None

    @staticmethod
    def cancel_flight(confirmation_id):...
```

Hotels APIs: Hilton

```python
class HiltonCustomerInfo:
    def __init__(self, name, passport, birthdate):...

class HiltonRoom:
    def __init__(self, room_type, available, price_per_night, date_from, date_to):...

class HiltonHotelAPI:
    @staticmethod
    def search_rooms(location, from_date, to_date, adults, children, needed_rooms):
        rooms = []
        rooms.append(HiltonRoom("Interior View", 6, 200.0, "29-01-2022", "10-02-2022"))
        rooms.append(HiltonRoom("City View", 3, 300.0, "29-01-2022", "10-02-2022"))
        rooms.append(HiltonRoom("Deluxe View", 8, 500.0, "29-01-2022", "10-02-2022"))
        return rooms

    @staticmethod
    def reserve_room(room: HiltonRoom, customers_info: list):...

    @staticmethod
    def cancel_room(confirmation_id):...
```

Payment APIs: Paypal

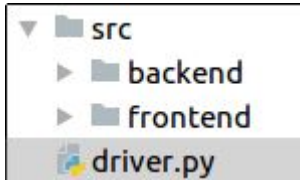
```
class PayPalCreditCard:
    def __init__(self, name = None, address= None,
                 id= None, expire_date= None, ccv= None):
        self.name = name
        self.address = address
        self.id = id
        self.expire_date = expire_date
        self.ccv = ccv

class PayPalOnlinePaymentAPI:
    def __init__(self, card_info : PayPalCreditCard = None):
        self.card_info = None

    def pay_money(self, money):...

    def cancel_money(self, transaction_id):...
```

Tip

- Although the whole project looks big, but consider:
  - We did **homework** about payment and reservation to prepare for the project
  - The idea of the different airlines & hotel is exactly the **same**. Once you implemented one of them properly, all others are matter of copy-paste
    - E.g. finish Air-Canada Flight part first properly
  - The goal of having these similar parts is to force you build **extensible** code
    - E.g. later we add Car and Cruise reservation
  - Another goal is to learn build common interfaces for close behaviours
  - And another goal to structure your code in several (sub)-packages
- Something missing? Make your own **assumptions**

Code Structure: High Level

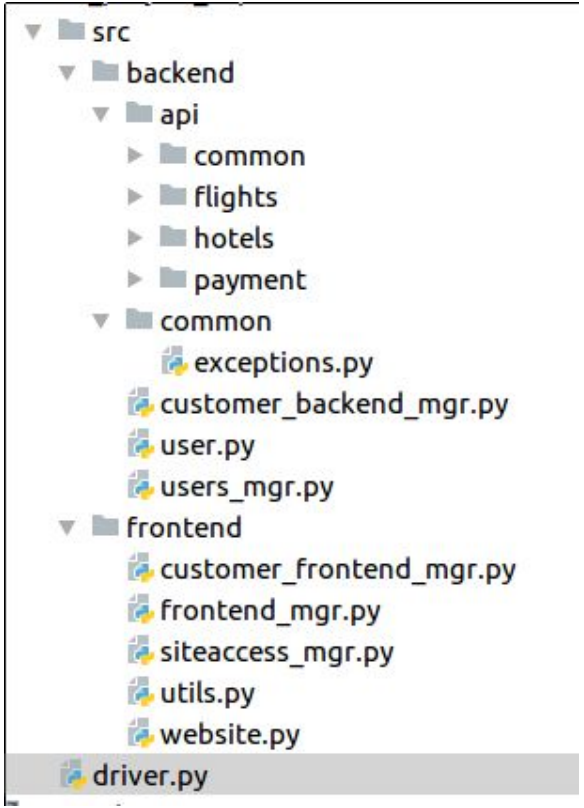
- Split your code to backend and front end
  - Backend: core logic. NEVER print to user from it
    - Think backend is a remote machine
    - User: using his own web browser or mobile
  - Frontend: Where use sees the screen
    - Menu options, selections and printing
- Next
  - Last slide: **optional**
  - My code overall structure




```
from src.frontend.frontend_mgr import FrontendManager

if __name__ == '__main__':
    # Local testing for this part
    mgr = FrontendManager()
    mgr.run()
```

Code Structure: Low Level

- You don’t have to follow that
- Don’t bother why such file names/split
- Just in case it is inpsiring



"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."