**Scenario:**
You are running an e-commerce business like Grofers. You have received **n** orders for the day. You have to deliver these **n** orders to the customers in 4 delivery slots ( 6-9, 9-13,16-19,19-23 hours). There are delivery partners who deliver these orders. Each delivery partner has their own vehicle and each vehicle has a carrying capacity i.e total weight of the orders which is dependent on the vehicle type (bike, scooter, or truck).

Bike capacity - 30 kg per trip
Scooter capacity  - 50 kg per trip
Truck capacity - 100 kg per trip

Each slot can take orders with an overall max weight of 100 kg. There are no trucks available in the morning slot (6-9 hours) and there are no scooters and bikes available in the evening slot (19-23 hours). All vehicle types are available for the other two slots (9-13,16-19 hours). You can't order more than 1 truck, 3 bikes, 2 scooters a day. Also, the vehicles' capacity should be used optimally.

**Requirements:**
1. Create a database with tables for storing information about the relevant entities as mentioned above.
2. Create an API endpoint that assigns orders to vehicle type based on the constraints mentioned above for a delivery slot to ensure all vehicle types are utilized during the day and no orders go undelivered.
3. Deploy the Application on heroku/ netlify/ aws/ digital ocean/ pythonanywhere / vercel
4. Share the links for documentation, code, and app link
    a. Documentation: Basic API documentation, some documentation wrt DB structure and logic used for assignment

**Submission Deadline:**  Two days from the assigned date

**Recommended:** Use of a python-based framework for the API creation.

**Brownie points:** For test coverage, REST API best practices, and coding conventions.

**Sample Case :**

You got 3 orders for the first slot.
For the first slot orders are:

```json
[{
    "order_id": 1,
    "order_weight": 30
 },
 {
    "order_id": 2,
    "order_weight": 10
 },
 {
    "order_id": 3,
    "order_weight": 20
 }
]
```

Since we can't assign orders to delivery partners who own trucks in the first slot.
The optimal solution, in this case, will be:

```json
[{
    "vehicle_type": "bike",
    "delivery_partner_id": 1,
    "list_order_ids_assigned": [1]
 },
 {
    "vehicle_type": "bike",
    "delivery_partner_id": 2,
    "list_order_ids_assigned": [2, 3]
 }
]
```