Python Developer

Logical:

1. Write a program that prompts the user to enter a credit card number as a long integer and Display whether that card is valid or invalid.Credit card numbers follow certain patterns.

   A credit card number must have between 13 and 16 digits. It must start with:

   - 4 for Visa cards (length 13)
   - 5 for Master cards (length 13)
   - 37 for American Express cards (length 16)
   - 6 for Discover cards (length 15)

   you can determine if a credit card number is (syntactically) valid as follows:

   - Multiply every other digit by 2, starting with the number's second-to-last digit, and then add those products' digits together.
   - Add the sum to the sum of the digits that weren't multiplied by 2.
   - If the total's last digit is 0 (or, put more formally, if the total modulo 10 is congruent to 0), the number is valid
   - Eg. 4003600000000014. Multiply each of the underlined digits by 2:
     1•2 + 0•2 + 0•2 + 0•2 + 0•2 + 6•2 + 0•2 + 4•2
     That gives us:
     2 + 0 + 0 + 0 + 0 + 12 + 0 + 8
     Now let's add those products' digits (i.e., not the products themselves) together:
     2 + 0 + 0 + 0 + 0 + 1 + 2 + 0 + 8 = 13. Now let's add that sum (13) to the sum of the digits that weren't multiplied by 2 (starting from the end):
     13 + 4 + 0 + 0 + 0 + 0 + 0 + 3 + 0 = 20. The last digit in that sum (20) is a 0, so the card is valid!

2. WAP to count the total votes and print the output in the form of a dictionary where the key is the candidate name and value is the total number of votes in the order of maximum votes.

   - write function for voting
   - vote takes a single argument, a string called name, representing the name of the candidate who was voted for.
   - If the name matches one of the names of the candidates in the election, then update that candidate's vote total to account for the new vote. The vote function in this case should return true to indicate a successful ballot.
   - If name does not match the name of any of the candidates in the election, no vote totals should change, and the vote function should return false to indicate an invalid ballot.

- You may assume that no two candidates will have the same name.

Write a print_winner function.

- The function should print out the name of the candidate who received the most votes in the election, and then print a newline.
- It is possible that the election could end in a tie if multiple candidates each have the maximum number of votes. In that case, you should output the names of each of the winning candidates, each on a separate line.

FastAPI:

1. WAP using FastAPI web framework using either MongoDb or any SQL
    - Create API to print “Hello World”
    - Create API for a School to perform CRUD operations for teachers or students or both. (School is the Database and teachers and students are 2 different tables)
    - Create API to assign students to a particular teacher
2. WAP using FastAPI web framework using either MongoDb or any SQL
    - Create User and login.
    - Create login user with authentication
    - Create login user with authentication and session timeout
3. WAP using FastAPI web framework
    - Create api to take two latitude and longitude and give distance between them using distance formula [$d=\sqrt{((x_2 – x_1)^2 + (y_2 – y_1)^2)}$]

SQL:

1. **Create query for result**

Table - routes

id	name	number
1	200-D	SBV-BHJ
2	300-A	ASD-WER

Table - route_points

id	route_id	order	distance
1	1	1	0
2	1	2	100
3	2	1	0
4	2	2	50
5	2	3	100

Expected Result

route_id	name	total_distance
2	300-D	150
1	200-A	100

2. **Make a query for get result for same route by stop id for source and destination**

Table - routes

id	name	number
1	200-D	SBV-BHJ
2	300-A	ASD-WER

Table - route_points

id	route_id	order	stop_id
1	1	1	1
2	1	2	2
3	2	1	1
4	2	2	2
5	2	3	3

Expected Result

route_id	source_stop_id	dest_stop_id
1	1	2
2	2	3

3. Make a query for result by station_id and slot

Table - station

id	name
1	Sta 1
2	Sta 2

Table - times

id	station_id	slot	time
1	1	1	6:30
2	1	2	6:45

3	1	1	6:40
4	1	2	6:50
5	2	1	8:10
6	2	2	8:20
7	2	1	8:15
8	2	2	8:30

Expected Result

station_id	station_name	slot	time
1	Sta 1	1	6:30
1	Sta 1	1	6:40