**Stage 1 Technical Interview**

**Question 1**

Our clients can refer their friends to join any of our events. They would share their email to identify the referral relationship.

Task: Use Flask to display the referral relationship for all clients under the main referrer. Implement a basic UI that is user friendly. Optional: Implementing a database.

A mock data set is provided in client.csv

Sample expected test cases:

|  |  |
| --- | --- |
| Input: | test@stripe.com |
| Output: | test@stripe.com |

|  |  |
| --- | --- |
| Input: | beemo@beemo.com |
| Output: | beemo@beemo.com  >>>> test@stripe.com |

|  |  |
| --- | --- |
| Input: | mainlevel@email.com |
| Output: | mainlevel@email.com  >>>> sublevel@email.com  >>>> sublevel2@email.com  >>>> user@email.com  >>>> Sublevel@email.com  >>>> User@email.com |

**Question 2**

Task: Build a simple API on Flask that accepts a POST request given an email address. Then, create a basic web page that uses JavaScript to create a POST request to retrieve the name and points given an email.

Sample Table:

|  |  |  |
| --- | --- | --- |
| email | name | points |
| string | String | int |

**Question 3**

Scenario: Our admin team would like to collect information about our client's subscription habits for business decision making.

Task: Share with us how would you implement the scenario - (you are free be as creative as you wish but robustness, practicality and simplicity would suffice)

Questions of interests:

- What would you implement? (Assuming a database and CRM of all the clients already exists)

- How would you implement? (Tech-stacks used, any special features, computation algorithms etc.)

- What kind of value would the admin team be able to get? (Hint: Tracking information is always useful)

**Question 4**

Scenario: We have several marketing platforms for marketing and sales. Each platform has their own version of storing client’s data.

Task: Suggest a plausible solution that would be best to store the data for use in the future and methods for data cleaning.

Question 5:

Scenario: A free finance news API allows 10 calls per minute. Each API call would take in a financial keyword (ie. inflation) and would take 5 seconds to process its sentiment analysis to be sent to the end-user. As the number of end-users increases, the frequency of calls increases greatly.

Task: Suggest one or more implementations that would allow the end-user to still receive the requested info with lesser disturbances or delay.

Question 6:

Scenario: As the project leader of an AI chat bot, your team informs you that the number of active users is reducing and the average use time of the users are decreasing.

Task: Suggest any features that can be added into the chat bot that would increase its subscription value (economic value or user experience). Try to maximize value while minimizing the cost (economic cost, computational cost).