Q.1. Tell us about your similarity calculation and why you chose it.
A.1. I used cosine similarity measure to identify similar users. Since, the data provided captures the user's online behavior and their interests which can be vary significantly from user to user. For example, one user can be highly active in finding out about certain types of courses and the other user, also interested in similar courses but not active as much and cosine similarity measures captures that relationship by using n-dimensional vector space and identifying the orientation of the user's activity vector with respect to other users. This is the reason I chose cosine similarity to calculate similarity distance between users.

Q.2. We have provided you with a relatively small sample of users. At true scale, the number of users, and their associated behavior, would be much larger. What considerations would you make to accommodate that?
A.2. The considerations I would make to accommodate the scale of the project are:

- I would try to change the distance calculation function to incorporate multi-processing to utilize as much hardware power.
- Run the distance calculation script on the server that hosts database and has at least 8 cores.
- Use open source cloud computing frameworks like Apache Spark or Hadoop MapReduce. Cloud computing will essential to handle the large scale of the project. Apache Spark would be the way to go due to better performance than Hadoop MapReduce.

And off course, I would use vectorized iterations instead of complex for/while loops.

Q.3. Given the context for which you might assume an API like this would be used, is there anything else you would think about? (e.g. other data you would like to collect).
A.3. Considering the scale of this project in production, API would not be very ideal for Iris to access data since there would considerable reliance on the network speed to send the request and receive the response from API. API would essential to query data on ad-hoc basis. But Iris should get the data from the data warehouse ideally located on the same server as Iris. The user similarity calculations should be done periodically and the output should be dumped into the database for Iris to make any recommendations.

About other data, I would like to add the location of the user to the similarity distance calculations.

Note:
The current API implementation as part of the technical interview uses SQlite database to store only user activity data and not the engineered features used to calculate similarity distance since SQlite has limitations on number of variables inserted in the table. So whenever the GET request is made to the API, the API performs calculations before sending the response which would take few seconds.