Api Task output :

A: When called directly, it should return the similarity among different candidates.

You can check the index.js file where I have implement this partA and mark it comment

else if(!uid2&&uid1){

let response = result.filter(item=>Object.keys(item)[0]==`user${uid1}`)[0];

res.status(200).json(response);

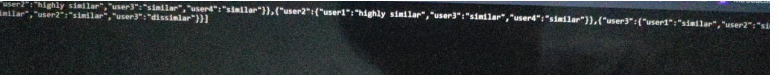
}





b. **Filters**: The API should be able to filter the results based on the people such that only the similarity between the responses of the passed candidate and everyone else is returned.

**{"user1":{"user2":"highly similar","user3":"similar","user4":"similar","user5":"similar","user6":"similar","user7":"similar","user8":"similar","user9":"similar","user10":"similar"}}**



**C.Pagination**

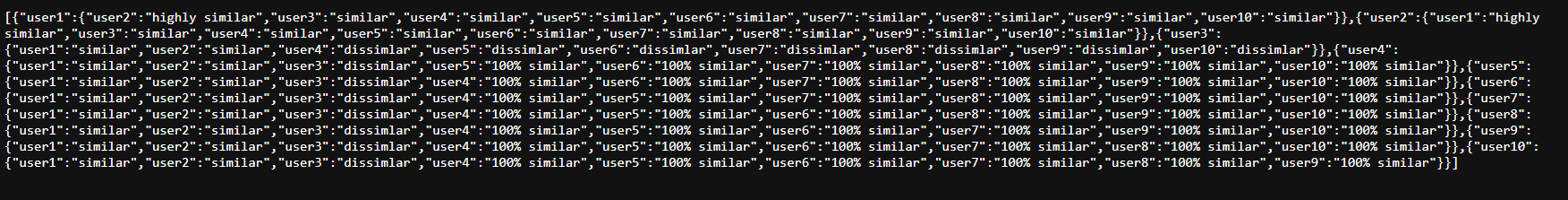
**[{"user1":{"user2":"highly similar","user3":"similar","user4":"similar","user5":"similar","user6":"similar","user7":"similar","user8":"similar","user9":"similar","user10":"similar"}},**

**{"user2":{"user1":"highly similar","user3":"similar","user4":"similar","user5":"similar","user6":"similar","user7":"similar","user8":"similar","user9":"similar","user10":"similar"}},**

**{"user3":{"user1":"similar","user2":"similar","user4":"dissimlar","user5":"dissimlar","user6":"dissimlar","user7":"dissimlar","user8":"dissimlar","user9":"dissimlar","user10":"dissimlar"}},**

**{"user4":{"user1":"similar","user2":"similar","user3":"dissimlar","user5":"100% similar","user6":"100% similar","user7":"100% similar","user8":"100% similar","user9":"100% similar","user10":"100% similar"}},**

**{"user5":{"user1":"similar","user2":"similar","user3":"dissimlar","user4":"100% similar","user6":"100% similar","user7":"100% similar","user8":"100% similar","user9":"100% similar","user10":"100% similar"}}]**



**And next page will display similary for user 6 , user 7 , user8, user 9 , user10**

**Searching :**

**Logic for this part is**

1. **Database Setup: assuming we have database where candidate response and information is present**
2. **API Endpoint for Search:**

**const express = require('express');**

**const router = express.Router();**

**router.get('/search', async (req, res) => {**

**try { const searchText = req.query.searchText.toLowerCase();**

**const candidates = await Candidate.find({ name: { $regex: searchText, $options: 'i' } });**

**res.json({ candidates });**

**} catch (error) {**

**console.error(error);**

**res.status(500).json({ error: 'Internal Server Error' });**

**}**

**});**

**module.exports = router;**

1. **Similarity Calculation:**
   * Inside the **/search** route, we can calculate the similarity between the search results and the user who initiated the search . we just reuse our logic for calculating the similarity
2. **API Integration:**

**const searchText = 'John'; // Replace with the user's search query**

**fetch(`/api/candidates/search?searchText=${searchText}`)**

**.then((response) => response.json())**

**.then((data) => {**

**console.log(data.candidates);**

**})**

**.catch((error) => {**

**console.error(error);**

**});**