

1 Statement Motivation

Abhishek Bhatia has just become the Overall Placement Coordinator of SPO, IIT Kanpur and he needs your help. Every year, Day1 of placements sees a huge havoc where the SPO has to manage a lot of interviews in an ad-hoc manner. There is confusion over which company/candidate is interested in which candidate/company and an un-documented control format that the SPO ends up adopting (to ensure maximum students get placed) becomes a challenging and cumbersome task for them. Fortunately some bright economists have already done a lot of the heavy-lifting for you.

1.1 Statement

Your goal is to make the placement process a lot easier for SPO (and our dearly beloved Bhatia). You need to develop a fully functional web-app that all students sitting for placements can log into. They should be able to indicate their preferences of the top 4 companies they wish to interview for. At the end of Day1 all companies will hand over their list of preferred candidates. Your portal should then take that input in and give the SPO admins a tentative list of placement pairs such that **maximum students are placed maximally-preferably** (with respect to their preference list). Assume that companies give a total ordering over all candidates they interview and the maximum number of candidates they want.

100pts.

2 Marking Breakup

If you have understood the problem statement above, you know that you need to be both **fast** and **optimal** in your back-end logic implementation so that the eager students get their Day1 results as soon as possible. Here is how we will evaluate you.

2.1 Dockerization

We do not care about your technology stack but setting your application up/moving it to production should be as simple as running **docker-compose up -d**. The quantity of marks you get in this section will depend on how easy it was for us to setup your application.

20pts.

Note:

- Full marks if it is as easy as **docker-compose up -d** or **./setup.sh**.
- You may assume that **docker** and **docker-compose** are already installed on the machine which will be used for testing.
- You can look at this project for inspiration.
- you will be marked out of 5 in this section if you do not use dockers at all.

2.2 Algorithm for matching - Sanity

Remember that there may not necessarily exist a stable matching. In such a case you need to ensure that maximum possible students get placed maximally. We assign you the task (and you may contact the SPO for help on this) on how to assign weights/develop a cost function to do this. Any function that penalizes on assigning a student a lesser preference will be accepted. Note:

15pts.

- There is no objective measure for full score in this section.
- You may assign different weights to different companies (intended minor technical glitch here: see if you get it).
- you will be marked 0 in this section if you do not use any sort of cost measure at all.
- You will get 5pts if you adopt any sane cost function (or weights).
- The remainder of the 5pts is left to our judgement (please add a **Rationale.md** in your submission explaining what you have done.)
- You will get 0 pts if you match some candidate/company incorrectly. Your algorithm however implemented must be sound.

2.3 Algorithm for matching - Speed

Since you have access to the internet (and in turn good matching algorithms), we will mark you on speed as well. Provided to you with this pdf is a json file for one such instance. Because all students and their parents and their relatives and the companies and basically everyone else on planet Earth are waiting eagerly for Day1 placement results the quicker your back-end can do it the better. Note:

15pts.

- We will not be simply giving full score to the fastest implementation. This is because a very fast implementation is to simply give out any random (valid) matching.
- We will rank the performance of all submissions on our test and score everyone accordingly.
- You will be marked out of 5pts in this section if running the test is any harder than doing either of the following `./runtest speedtest.json` or making a POST request to a special **api** endpoint called `../api/speedtest`.
- You will get 0pts if you match some candidate/company incorrectly. Your algorithm however implemented must be sound.

2.4 User Interface

Okay guys, lets be honest Bhatia isn't really a savage (even though a lot of us would like to disagree). No one likes to see very badly formatted and un-intuitive UIs. We therefore reserve the right to subjectively score you out of 20 for the User Interface you come up with. Note:

20pts

- You will be marked out of 5pts if you do not use jQuery OR any front-end framework.
- You will be marked out of 15pts if you use only jQuery AND no front-end framework.
- You will be marked out of 20pts if you use a front-end framework.

2.5 Deployment

Well if you end up deploying your project on heroku or aws, thats a good plus. So you deserve points for that. Note:

10pts

- You will be given 0 pts if you simply deploy any utter crap of a software. If you are deploying it, we assume it will be fully functional.

2.6 Testing

Good software should be well tested. We will therefore give 2pts (upto a maximum of 10) for every test you write. We should be able to run those tests conveniently using a script. Note:

10pts

- You will be given 0 pts if you simply make a folder called **tests** and do not explain how we can run them ourselves. Typically, the last step of your dockerfile should be running those tests.

2.7 Admin & Company interference

On the crucial Day1, the SPO might need to make some ad-hoc adjustments. E.g., Company A might suddenly like Aditya Mishra or Abhyudhay Pandey so much that they contact the SPO even before Day1 is over and ask them to make a speedy offer. In such a case the SPO should have an option (on their admin portal) to manually add this match before even all the Day1 lists are in and they run the **generate list** (which you also need to mandatorily implement) button. Also, say some candidate is done with his 4 interviews surprisingly fast (we all know why) and the SPO decides that getting this person a 5th interview is not the worst idea. They should be able to do so from their interface.

10pts

The company interface should allow them to upload their list of students OR "Request a special student" from the SPO (as described above). Note:

- You will be given full score if these are implemented.

2.8 Github

An integral part of development is using git/github wisely. We will therefore award upto 15 bonus points if we see that your github history is like that of an actual repository. (People making branches, PRs, pushing from time to time, and opening issues). Note:

- The quantity of bonus is subjective.

3 Submission

Feel free to ask questions about how to do this project (either from us or from the SPO). We will have a messenger group specifically for asking doubts. The submission will be through a google form where you will submit links to your GitHub repos. You should make your repositories private and invite all PClub coordinators as collaborators.

P.S. - Feel free to point out errors!

15pts, bonus