Task – Total time 5 hours

1. Use the available data from a file (data is provided as a Json file) to save it in a database (Quest DB recommended but you can use any database which is suitable) in real time based on the difference of the timestamps in the file. This should act like a publisher where the the published data based on the timestamp should automatically update the database with all the entries. This should be done with the help of an API. You can decide and use the API framework of your choice. You can decide format of the saved data on the database. Description of the data is shared below at the end of this task description.
2. Prepare a simple frontend page with two components stacked vertically using any library you want.
    a. 1st component
        i. First component on the top is a graph where x-axis should be time and y component should be a value from the saved data. The displayed value on the graph should depend on a drop down menu or any other tool to select one of the multiple options from the database and change the graph as soon as the option is changed on the drop down tool:
            1. Number of humans at that time
            2. X position of human
        ii. The averaging of the data shown on the graph should be done automatically (preferably by the database during fetching from database) depending on the selected duration of the data.
    b. 2nd component
        i. The second component should be stacked vertically below the first component and should be a positional heatmap using the x and y position from the database and intensity is the value of the selected option from the graph of first component (You can use heatmap.js library or other library)
        ii. The second component should show heatmap of the data during the timeframe seen that time on the graph (first component). You can assume all the parameters for the heatmaps if not stated here.
3. You should use **React for frontend**
4. **If something is not mentioned here, please assume it by yourself and try to give the best solution as much possible.**
5. Prepare a short description of your work. Mention something that you think is important for someone else to understand. Mention anything that you think you have done innovatively and why you did so. This description should be part of Readme on Github Repo.
6. Use a **Github Repo** to work on this task.
7. **The Readme file should contain all the steps to install any dependencies or use your work even for a layman**. The commit history should show the changes you have done during the development process.
8. **BONUS TASK** – If you have some time left after implementing all the features above then Dockerize the whole frontend and backend part using Docker which should be able to run using just a single command like **docker-compose –up**
9. **BONUS TASK** – Describe in a separate file how all this can be implemented on AWS. What all services will be used from the available list of services by AWS.
10. **If completing this task is not possible in the given time, prioritise the components, work, and give the best solution possible with a compromise between features.**

Data Description:

The Json file provided to you is a list of dictionaries.

Each dictionary has a timestamp which is a unix timestamp which should be used as a timestamp always to save, retrieve and display data on the frontend.

There are various instances in that timestamp. Each instance represents a human and its x and y positions. The IDs of the human are also mentioned in instances and should be considered as the same person.

The data looks like –