

About TensorGo and Expectations

TensorGo is a product based company that is working on the most advanced technologies related to Computer Vision at edge.

This role requires working with one of the smartest minds in the industry by contributing your innovative ideas and experiencing yourself from the customers vision to make their experience delightful. We expect our teams at all levels to come up with their own thought process and Innovation in making us getting better everyday in the world we live in.

It is certainly important to showcase yourself in this assignment, how you can be different from your general responsibilities and create a dent to the world of Innovation from our research labs.

TensorGo team wishes you Good Luck!

Instructions

1. Duration 4 Hours.
2. Once you are done with the assignment, please take a screenshot of the output and send your work to HR by email.
3. Any further questions you might have to be assumed as per your thought process. Please send Readme.txt file which contains Your understanding of the Scope, assumptions and installation steps to launch the application.

Use Case

1. Access user API from <https://gorest.co.in/> i.e <https://gorest.co.in/public-api/users> [Please login with your Gmail ID to get the API Token]
2. You will get the result as following:

```
"result": [  
  {  
    "Id":18,  
    "name":"Aditeya Arora",  
    "email":"aditeya_arora@abc.com",  
    "gender":"Female",  
    "status":"Inactive",  
    "created_at":"2020-09-07T03:50:03.766+05:30",  
    "updated_at":"2020-09-07T03:50:03.766+05:30"  
  }  
]
```

Note: Object field values should go into the User Master table in Database

3. Providing an edit screen for the above data in the frontend and edited data should be updated in the DB, note fetching the DB and updating the data in DB should be done by Web Service.
4. Create a Export script to export the User Master data a in a single sheet CSV file.

What Do We Expect From You?

1. MicroServices to fetch the API and store the data in the Database, Publish the User data to the frontend and update the data in the table [3 MicroServices should be written]
2. FrontEnd development to view and update the user data
3. Microservice to export the data into CSV
4. Readme.txt
5. DB Script
6. Frontend webpage screenshots
7. API access postman screenshots and Postman collection file

Technology stack

FrontEnd : React JS

WebService/MicroService : NodeJS

Database: NoSQL

1. MicroServices should be developed in **NodeJS** for GET, PUT or POST
2. CSV Export Microservice should be developed by **NodeJS**
3. Data should be stored and fetched from **NoSQL**
4. **React JS** Framework should be used for Frontend development