

**Note: Please don't share this document with anyone!**

We are looking for people who can build, maintain and scale crud applications using ReactJS and Python for a telecommunication backend. And the candidate should also have the attitude to pick up knowledge such as networking or a new full stack language and framework in a short period of time. To find the right candidate we have set up the following test.

## Overview

You will build a simple web application with ReactJS as Frontend, Python as backend and host it in a cloud. Completing all steps is not necessary but completing each additional step demonstrates how good your problem solving skill is. **Please read the assessment criteria.**

Send us an email if you have a hard time understanding the tasks.

## Assessment Criteria

- The primary assessment criteria is your learning and adapting capability. Some of you have already worked with ReactJS & python and have some experience, others have programming experience but no React/Python experience.
- If you don't have a web dev background and can self learn basic concepts and build an application within a short period of time, we will value that immensely and you will score high.
- If you already have web dev experience, we will value intermediate usage of the tools which will demonstrate your deep understanding of the tools.
- The earlier you submit with the more features, the more score you will get.
- Clean & Efficient Code will score high. Early submission will score high as well. Please note that completion of the task is scored higher than early submission, so try to complete as many steps as you can within stipulated time.

## Tasks

- Build a basic web app with **ReactJS** and Python with [this json](#), make a table visualization with the data in the home page of the website. [Here is the csv](#) of the same data if you need it. It is not absolutely necessary to build the backend in python, other backends will do as well but will score lower, but **you must make the frontend with ReactJS.**
- Load the data in an sql server, and make the frontend table rows editable. We want to see if you can make a crud application. If you have moved into this step, you are switching your model from the json to the sql server, keep the json and sql models separate, and use the new sql model from this step and onward. If you can make a git version for the previous step named jsonModel, and starting from this step make the new

version sqlModel, it will be easier for us to go back and forth between the different models and test it out.

- Add a line chart above the table visualization, put the close column in the y axis and sorted date in the x axis. You can also add the volume column in the y axis by making it a bar chart, and accommodate the line and bar chart together by making it a multi axis chart. Add a drop down in the chart so that we can choose the trade\_code column which will change the data in the line chart.
- Make any other kind of visualization that you can think about, be creative.
- If you completed any of the steps above, publish the code in a folder, upload the code and application in github, add a readme.md in the repository in which explain what did you learn from this project and what challenges did you face completing it, then mail the github link to us at [careers@janatawafi.com](mailto:careers@janatawafi.com), make sure to mention your name and phone number.
- Deploy the application in a cloud hosting service. You should find a lot of free hosting if you just search for it. You can use pythonanywhere or render for example. Share us the link to the app.
- Do send us a mail if you get confused by anything, we will explain.