NORTH PARK INNOVATIONS

NORTH PARK INNOVATIONS GROUP INC.
SOFTWARE ENGINEERING INTERNSHIP
TAKE HOME TEST

This is an open-end test so feel free to be innovative and show your skills if you would like to add something more to enhance the solution for below requirements. Better you pitch yourself, better are your chances!!

Objective:
Develop Single Page Application (Ticketing System) using HTML, CSS, JavaScript and VueJS.

Development Guidelines:

1. **Deadline is 04/22/2019 9AM EST.**
2. **Use of Google, MSDN, StackOverflow, GitHub, etc. is allowed.** Feel free to go through any of tutorials online and use the code explained there.
3. **VueJS should be used as the JavaScript framework.**
4. **jQuery should not be used!!**
5. Any CSS library like **Bootstrap/Bulma/Animate, etc.** could be used.
6. Frameworks like Express, Webpack, Nuxt, NodeJS, etc. could be used.
7. How to use Vue: **https://vuejs.org/v2/guide/index.html**
8. **There is no right or wrong answer for this task. If you are stuck or confused with any requirement, you could proceed in any direction (use your judgement) and explain that in the document**.
9. Try to understand features like two-way binding, v-if, v-else, v-for, v-show, v-model etc. which will really help in completing this task.

Task:

1. Create a login screen (Screen 1) which would authenticate the user id and password from provided JSON file (User.json).
2. If successfully authenticated, Ticketing Dashboard (Screen 2) will be visible, and Screen 1 will disappear.
3. On Screen 2, some summarizing facts/charts about the ticketing data available should be shown on the top. Just below summary, tickets and their info (from Sample Data.json) should be displayed in **optimized manner**. Show only 8 out of 11 fields (like Requestor/ITOwner) which you find to be most relevant in the Dashboard.
4. **There should be a feature to search and sort tickets based on user input.**
5. If a ticket row is clicked, open a pop-up/modal which shows all the fields (Requestor, ITOwner, etc.) described in Sample Data.json.
6. Try to make UI as responsive as possible.
7. UI should look good so use any CSS library if required.
8. Application should be smart enough to tackle edge-cases.

Submission Guidelines:

1. Please create a document with number of hours spent on this task and distribution of hours along with explanation.
2. Upload the final solution with all the files on Github/Google drive and share the link for it while replying to this email.

Data Definition of Sample Data.json:

Requestor	Employee ID who submitted the ticket
ITOwner	Employee ID of IT employee who serviced ticket
FiledAgainst	Functional area the ticket was filed
Severity	Submitter assigned severity of ticket
Priority	IT assigned priority of ticket