Junior Sales Engineer Exercise

As a Junior Sales Engineer at a growth stage startup you’ll own technical relationships with accounts from pre-sales all the way through ongoing support. This involves answering tier 1:2 support issues, troubleshooting tough technical problems, and working with core engineering to diagnose and fix bugs.

The goal of this exercise is to (1) understand your troubleshooting approach, (2) understand your communication style, (3) assess your technical aptitude from both a coding + systems standpoint, and (4) assess your ability to operate with limited information.

Part I: Troubleshooting

Please provide a brief answer to the following questions. Goal here is to understand your troubleshooting technique as well as ability to communicate technical issues in written form.

1. Describe a difficult technical problem you helped resolve for a customer. How did you troubleshoot the issue? What considerations did you have to take into account?

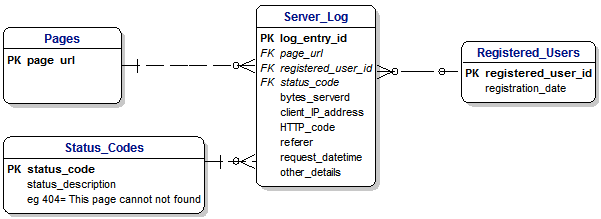
***TODO****: provide answer*

1. Describe a system or process you helped design. What are its best design characteristics? What tradeoffs did you have to make to complete the project?

***TODO****: provide answer*

Part II: SQL

A key part of troubleshooting issues is running effective SQL queries to analyze customer traffic logs.



Based on the DB schema above please write queries to answer the following:

* Display all requests from IP 192.225.213.20 to page ‘/admin’ on 4/25/14.

***TODO****: provide SQL*

* Which pages return the most HTTP 500 status codes? How many?

***TODO****: provide SQL*

Part III: Scripting

In language of your choice (bash, python, ruby, etc), write a command line script that does the following:

* Queries WMATA public API and outputs the next Orange line train arriving at ‘Ballston’ metro stop.
  + <https://developer.wmata.com/docs/services>
  + API Key: kfgpmgvfgacx98de9q3xazww
* Please upload repo or gist to Github.

***TODO****: provide github*

Part IV: NGINX

10Pearls relies heavily on NGINX. During this part of the exercise you’ll build a simple web server that will take requests, filter for mobile vs desktop access, and return correctly formatted results to the client.

Node guidelines:

* Provision an Ubuntu based EC2 instance on AWS (free tier is fine)
* Install and configure NGINX to meet the following requirements:
  + Display mobile index for mobile devices

Feel free to reference any material you find useful, Google is your friend.

***TODO****:* **EC2 web address**

***TODO****: Github NGINX configuration file*

Good luck!