Software Engineer - Performance Task 2

We are excited to have you complete the next step in our Engineering hiring process. This performance task is intended to provide you with an opportunity to demonstrate your potential to take on the responsibilities of this role while providing our hiring team with insight into how you might approach the type of activities that this person will regularly engage in.

We want to be respectful of your time. Please take **no more than 3 hours** to complete Part 1 and *either* Option 1 or 2 under Part 2 below, and **email your submission to eng-recruiting@achievementnetwork.org within the timeline communicated to you via email.**

If you complete one part **-** either Part 1 or Part 2 **-** in less than three hours, you may choose to work on the other part in time remaining and to include partial results in your submission, along with a note indicating how much time you spent on each part of the task.

If you feel you have questions about this activity that prevent you from completing it, please don't hesitate to email eng-recruiting@achievementnetwork.org.

Step 1

Submit a sample of your work that illustrates software you have designed and implemented to support either system infrastructure or application functionality on the front end, back end, or both. Provide some context regarding the problem you were trying to solve and its impact on your users. Walk us through the frameworks or technology selected, why those were the best fit for the project. Highlight an example of something you learned while implementing this code.

Step 2

Choose either the front-end *or* the back-end focused task below. You may need to make assumptions in order to complete that task. Please state those assumptions in your submission.

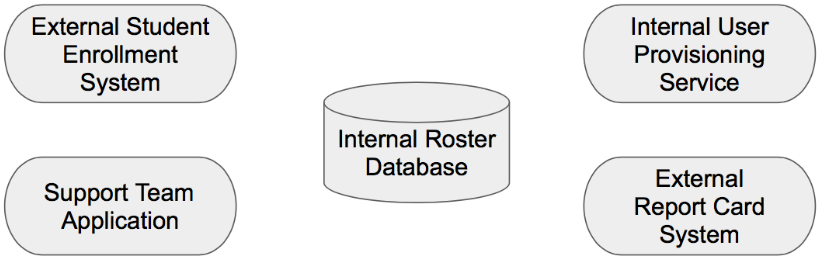
Option1: For front-end developers:

Review the API provided by https://github.com/Hipo/university-domains-list-api. Consider how you would implement an application that calls the public API to search for and display a list of universities that meet user-specified filter criteria. The application should allow users to update filter criteria and sort the list by all fields, including university name, country, state/province, domains, and web pages.

1. Describe the approach you would you take to designing a solution, including design challenges and questions that come to mind and your strategies for resolving them.
2. Using diagrams and/or text, whichever you prefer, outline what your high-level design would look like, including any technology choices you would make to deliver a solution.
3. Assume http://universities.hipolabs.com also supports an API to add new records to their database. Describe how you would extend your application to allow users to enter and save data for a new university by calling that API. Again, explain any questions, decisions, and technology choices.
4. Share your thoughts about additional work that you would consider doing before releasing such an application for production use.

Option 2: For back-end developers:

Student enrollment data maps students to each of their classes at a specific school. This data is updated in bulk at the beginning of each school year, then intermittently updated as students join and leave classes and schools during the year. The ecosystem for managing student enrollment data and their software services includes::


External Student Enrollment System
Support Team Application
Internal Roster Database
Internal User Provisioning Service
External Report Card System

- The External Student Enrollment System (ESES) - a third-party system that large schools use to manage student enrollment data. The ESES runs a job every 10 minutes to send enrollment updates to an internal service that updates the Roster Database.
- The Support Team Application (STA) - allows internal product support team members to submit enrollment updates at smaller schools to the Roster Database.
- The User Provisioning System - responsible for assigning unique user ids to new students and creating/managing student accounts so students may access multiple applications, including the Report Card System.
- The external Report Card System (RCS) - consumes student enrollment information. The RCS allows students to view their test scores and report card data.

Consider how you would deliver a system that processes incoming enrollment data from the ESES and STA, and provisions new students in the system so students can access their grades in the RCS.

1. Describe the approach you would you take to designing a solution, including design challenges and questions that come to mind and your strategies for resolving them.
2. Using diagrams and/or text, whichever you prefer, outline what your high-level design would look like, including any technology choices you would make to deliver a solution.
3. Share your thoughts about additional work that you would consider doing before releasing your solution for production use.