

Full Stack Developer Code Challenge

**Scenario Overview:**

In this challenge, we'd like you to step into the shoes of a Senior Full-Stack Developer at NewYorker. Imagine that your first task on the job is to review the code written by our junior developers. The code has some areas that need improvement.

Your role will involve not only identifying issues in the code but also sharing constructive feedback and solutions to help them improve their coding skills.

**The Challenge:**

The code written by our junior developers: [GitHub - dthagard/salesforce\_backup: Performs automated backups of SalesForce.com data.](https://github.com/dthagard/salesforce_backup)

**Tasks:**

**Task 1**:

Your task is to thoroughly review the codebase and identify any issues related to code quality, readability, maintainability, or best practices.

**Task 2**:

Explain how you can address these issues and propose solutions. Assume that you will need to mentor the junior developers and guide them toward implementing the suggested improvements. So, you don't need to fix the code yourself but should suggest ways to address the problems.

**Task 3**:

We'd like to transition the existing command-line application to a modern web application. Please provide your insights on the following:

* **Overall Solution**: Provide an overview of what the overall solution would look like. Which solution components do you identify? What technologies, frameworks, and tools would you recommend using? How would you ensure that the web application maintains or improves upon the functionality of the existing command-line application?
* **Frontend Structure**: Explain how you would structure the frontend of the web application. What libraries or frameworks would you use, and why? Discuss considerations related to user interface design, user experience, and responsiveness.
* **Backend Structure:** Outline your approach for structuring the backend of the web application. Which programming languages, frameworks, and databases would you recommend? Address aspects like data storage, authentication, and API design.

**Challenge Guidelines:**

* Please note that we do not expect you to implement this application. Instead, focus on preparing your ideas so that you can effectively present them to the development team.
* Feel free to include any diagrams, charts, or mock-ups to illustrate your ideas if you find it helpful.
* Please submit your feedback and recommendations in a written report or document. You can use plain text, markdown, or a common document format (e.g., PDF).
* You will have one week to complete this challenge.