Velozient™

Coding Challenge

Password Manager

2022

Summary *3*
Password Management *3*
Algorithm & Approach *4*
Best Practices *4*
Bonus Points *4*
Technical Dependencies and Libraries *4*

Summary

Velozient connects accomplished LATAM software engineers with North American software organizations.

During the interview process, Velozient clients like to see a coding example so they can understand software development styles and skills. This Password Management coding challenge intends to provide such an example and should take you just a few hours to complete.

Please read the instructions carefully and make sure you submit your written "Algorithm & Approach" in addition to the coded solution.

If you have questions about the assignment, please ask.

Please utilize React.js, Node.js and C# related technologies to develop your solution.

Password Management

We all have a large number of usernames and passwords that we must remember. And each site/app has a different password policy and sometimes they assign our usernames to us so we have varied usernames and passwords.

And for the apps/sites we only use occasionally, we can constantly be forced to go through a "forgot your password" and sometimes "forgot your username" process because we simply cannot recall what they are.

We would like you to create a very minimal password management application so we can see how you code and how you think.

Of course, we aren't asking you to develop a market-ready, feature complete, password management solution like LastPass. But let's utilize LastPass as an example of what we'd like you to do - primarily in terms of "cards" and "card contents."

Here is an overview of LastPass: https://www.youtube.com/watch?v=a86PbT5XZt8

- Please review these two things in the video to keep this simple:
  1. ***At 26 seconds into the video*** – note the cards on the screen - one card for each application or website with the user specified name visible.
  2. **At 30 seconds** - card contents: URL, Name, Folder, Username, Password, and Notes.
- We would like you to limit the card contents to:
  - **URL:** will be typed in or cut/pasted into the field
  - **Name:** whatever the user wants to put for that application or site
  - **Username** and **Password** are self-explanatory

Your user interface should:

- Present "cards" that tile across the screen and allow the user to scroll down through them.
- Provide the ability to create new cards. Notice that URL, Name, Username and Password must be required.

- Provide the ability to edit cards.
- Provide the ability to delete cards.
- For the Password field, provide support for it to be obscured or unobscured. And provide the ability to copy the password into the clipboard so it may be pasted by the user into a password field when signing into an application of website.
- Provide a very simple search/filtering – use only the Name field.
- Provide the following REST API endpoints:

- /password-cards
    - GET Get all "cards"
    - POST Creation of a new "card"
- /password-cards/{id}
    - PUT Editing of a specific "card"
    - DELETE Removing of a specific "card"

Don't use a database. Please store data in memory, in a single file or serialize objects as needed. There is no need to retain anything afterward.

Algorithm & Approach

Please provide a written summary (just a few sentences or paragraph) describing your architecture/approach and how you decided to break apart various components and why. Please tell us about any assumptions you made and any limitations of your solutions are and why you chose those.

Best Practices

Be sure that your solution is well documented and follows best practices. Be sure that your code is readable and easy to follow.

1. Write as few lines as possible of legible code.
2. Segment blocks of code in the same section.
3. Use indentation to mark the beginning and end of control structures, methods, and/or functions used and developed throughout the project.
4. Write portable code. That way it will work on any environment.

Bonus Points

Using Python, GraphQL, Selenium, Postman and JMeter or similar technologies will be a benefit to you if you secure this job.

Technical Dependencies and Libraries

Please add or explain any dependencies and/or libraries that you are using in your solution. Also, do not forget to add the framework versions and applications you are using to write, compile, and execute your code (i.e., Visual Code or Microsoft Visual Studio).