Brittany Bolling-Nilsen

Final Report

"Junior Dev Interview Prep" is an app designed to help junior backend developers prepare for interviews. The main purpose of this app is to let junior backend developers practice answering interview questions and practice solving coding challenges they could face during a technical interview. There is a page with flashcards so users can practice answering questions. The flashcard sets/flashcards created by admin users are default sets/flashcards that users can practice with. Users can use (not edit or delete) the default sets/flashcards and create their own personal sets/flashcards. The flashcards must be saved to sets. For example, "Programming Foundations questions" and "Common Interview Questions." There is also a page where they can solve small tasks (for example, the FizzBuzz problem). First, the user will choose a category (for example: loops, maps, sorting). They will be given a challenge prompt (for example, the FizzBuzz prompt), then the user will have to enter the code to solve the problem. If the user needs help, they can see a hint. The user can also choose to see the solution. For certain problems, the users can click "Better Solution" to see a better way of solving the challenge. Users will be able to create their own profiles, reset their passwords, and have their own study sets. This project will help junior developers feel well prepared for their interviews, help junior developers continuously practice and perfect solving basic programming challenges, and keep junior developers motivated. There are multiple possibilities to add additional functionalities to this application, for example sharing study sets amongst users, checking if their code solution was the same as the written solution, practice projects, and much more.

While working through the project, I was faced with multiple challenges, for example: implementing a secure code editor in the application, implementing a secure authentication system (especially on the client side), and fetching all the admin's flashcard sets/flashcards and the logged in user's personal sets/flashcards. I learned eval() is evil and is a major security concern, authentication is much more complicated than I once thought, and how to implement pagination/filtering in a real-world project. The client side is not as secure as it should be, and this is something I will need to fix after submitting my project. It is currently functional, but does propose a few issues such as:

- the user is not logged out or notified when their JWT token is expired
- the token is being saved in localstorage for now
- if a user is not logged in they will see the flashcard sets/code challenge category paged briefly before being redirected to the login

Overall the customer's main requests have been met and are satisfied with how the project is turning out.