Interview Session Summary

Session Information

Candidate

Oleksandr Dubas

Company

Globaldev

Job Posting

Seniority

Middle

Interview Summary

Total Score

82/100

Status

Completed

Suitability by AI

Yes

Questions AI Evaluation:

INTRODUCTION DIALOGUE:

QUESTION :

Question:
Describe your experience with Webpack and how it can be used to manage and optimize assets in a front-end project.

DIALOGUE:

AI Interviewer
Describe your experience with Webpack and how it can be used to manage and optimize assets in a front-end project.

Candidate
Generally, Webpack allows us to pack all the code, all the libraries of our project and minimize it to make the app more efficient. Also, Webpack allows us to do tree shaking when we build our project. It removes the unused chunks of code which take up space and are not useful in production. Also, Webpack is useful when working with images, for example. Using it, we can efficiently use images. It's just a general tool for managing all our code and libraries and using assets, etc.

AI Interviewer
That's a good starting point! Could you give a more detailed answer or share a concrete example?

Candidate
Yeah, for example, we have a folder with our images and using Webpack we can use those images, and they can be stored more efficiently. For example, Webpack can be used to work with CSS preprocessors. Webpack is responsible for transpiling those preprocessors to plain CSS. So basically, Webpack allows us to work on a higher level of abstraction.

Score:
70

Explanation:
The candidate's response demonstrates a basic understanding of Webpack's functionality and its role in managing and optimizing assets in front-end projects. They mention key features such as code minification, tree shaking, and working with images and CSS preprocessors. However, the explanation lacks depth and detailed

examples of how these features are implemented in real-world projects, which would be expected at a middle level. Overall, while the answer is correct, it could benefit from more practical detail and concrete examples to elevate it to a higher score.

QUESTION :

Question:

How do you typically implement a component in React and what lifecycle methods do you use most frequently?

DIALOGUE:

AI Interviewer

How do you typically implement a component in React and what lifecycle methods do you use most frequently?

Candidate

So first of all, a component in React is just a function but it uses JSX special syntaxes, which then transpiles JSX in a component in a simple function. Usually, a component has props, special read-only inputs for the component which allows us to get some data. A React component should return a React element. For lifecycle methods, if we're talking about functional components, the most popular lifecycle method is useEffect. UseEffect allows us to run the code when the component mounts, when it updates, and when it unmounts. The first attribute of the useEffect is a function which will be run and the second is the optional parameter of the dependency array. The useEffect function will run when one of the properties of the dependency array changes. We can return another function, and this code will run when the component unmounts. This allows us to stop subscriptions if we had them or run cleanup functions.

Score:

85

Explanation:

The candidate provided a good overview of how to implement a component in React, effectively explaining both the basic structure and the use of JSX. They correctly highlighted 'props' as read-only inputs and discussed the return of a React element. For lifecycle methods, they focused on 'useEffect', which is relevant for functional components, and explained its usage well with clear details about its parameters and functionality. However, the answer could have included more depth, such as mentioning class component lifecycle methods like 'componentDidMount' or 'componentWillUnmount' for contrast, or providing specific examples of scenarios where 'useEffect' would be particularly useful. Overall, the response was clear, correct, and well-structured, earning a high score.

QUESTION :

Question:

What are some strategies you employ to optimize the performance and loading times of a complex React application?

DIALOGUE:

AI Interviewer

Thank you for that detailed explanation. What are some strategies you employ to optimize the performance and loading times of a complex React application?

Candidate

First of all and most important is React component composition itself; it should be smart. Sometimes when we put a child component that does not have any props passed or does not depend, so to say, on the parent component, it still will be rendering when the parent component changes. So the most important is component composition to not cause many renders for the app. We can use useMemo hook which allows us to store the result of complex calculations so not to run the function which requires a lot of work every time the component renders. Also, we can use useCallback which allows us not to build a new function every time the component renders if this function doesn't take any props. We can use React.memo to memorize components. This way they won't be rendered when their props are unchanged. We can talk about lazy loading, loading images, and chunks of code only when they're directly used. One more important thing is working with lists of data like mapping components, we should use keys and those keys shouldn't be array indexes, they should be unique and unchangeable, so React knows what component has been changed and only renders those components that have been changed. So in conclusion, the main things are component compositions, working with lists, using React.memo, useMemo and useCallback, and also lazy loading.

Score:

90

Explanation:

The candidate's answer demonstrates a strong understanding of performance optimization strategies in React applications. They provide multiple correct techniques, such as component composition, the use of `useMemo` and `useCallback` hooks, and `React.memo`. Additionally, they highlight the importance of unique keys in lists, which is a crucial point in React performance. The response is well-structured, covers a variety of strategies, and includes relevant technical details, making it suitable for a middle-level candidate.

QUESTION :

Question:
How do you use CSS preprocessors or CSS modules to enhance the maintainability of your stylesheets?

DIALOGUE:

AI Interviewer
Great insights! How do you use CSS preprocessors or CSS modules to enhance the maintainability of your stylesheets?

Candidate
Yes, obviously CSS preprocessors are in general a more efficient way to use CSS. We can reuse classes, meaning we can put classes inside classes without the need to rewrite some component data. Basically, CSS preprocessors allow us to write less CSS and to structure our CSS stylesheet more intuitively. For example, when I have a section and inside this section there is a division, and inside this division, for example, there is an anchor tag. In simple CSS we would write different classes for those tags or long selectors. Using CSS preprocessors, we can just put the division block inside the section block and this way we would basically write the same as writing long selectors in simple CSS. We can also talk about different mixins and more flexible variable usage in CSS preprocessors, as well as more convenient media queries.

Score:
83

Explanation:
The candidate provides a solid understanding of how CSS preprocessors enhance maintainability. They correctly mention class reuse, structure, and specific features like mixins and variables, which demonstrate a good grasp of the topic. However, while the answer is correct and includes relevant examples, it lacks a bit of depth in exploring the practical benefits or contexts where these features have been advantageous in real projects. Overall, it's a strong response for a middle-level candidate.

QUESTION :

Question:
Describe an experience where you had to resolve a cross-browser compatibility issue. What steps did you take?

DIALOGUE:

AI Interviewer
Thank you for your explanation. Describe an experience where you had to resolve a cross-browser compatibility issue. What steps did you take?

Candidate
In my experience, I had a lot of troubles with cross-browser compatibilities. I'll explain one example. I had an issue where some JavaScript code wasn't working on mobile Safari browsers. In that particular case, I had to look up compatibility of the specific method I used in the code and basically found out that it was not compatible with Safari mobile browsers. I had to come up with another approach using methods compatible with both Safari and other browsers. The simplest way is just to look up the compatibility of specific functions or methods in JavaScript and use only those that are compatible everywhere to avoid issues.

Score:
80

Explanation:
The candidate's answer is correct, and they clearly describe a specific example involving mobile Safari, which adds relevance and context. They outline the steps they took, such as researching compatibility issues and adapting the code accordingly. However, the response could benefit from more depth, such as detailing the methods they ultimately used, the tools for checking compatibility, or any testing strategies employed. Overall, it's a solid answer but lacks some practical detail that would elevate it to a 'Very Good' level.

QUESTION :

Question:
Can you explain the main differences between ES5 and ES6 in JavaScript, and why you might choose one over the other in a project?

DIALOGUE:

AI Interviewer
Thank you for sharing those examples. Can you explain the main differences between ES5 and ES6 in JavaScript, and why you might choose one over the other in a project?

Candidate

ES6 is basically a newer standard for ECMAScript, which JavaScript is based on. ES6 came out 10 years ago, so I would say it's pretty popular and used right now. I don't see a reason to use ES5 instead of ES6 unless we're talking about very broad browser compatibility. I hardly see any situations like that, maybe when the project initially was written in ES5 and it's very large, it would take a lot of time to rewrite it in ES6. A lot of differences include arrow functions, the spread operator, and classes were introduced. ES6 allows us to write code which is more easily comprehensible. For example, spread operator is very convenient for shallow copying objects and arrays. It's just more convenient to use ES6. It's like syntactic sugar where we can use classes without function constructors. Maybe do you have some specific questions about ES6?

Score:

83

Explanation:

The candidate provides a solid overview of the key differences between ES5 and ES6, highlighting major features like arrow functions, the spread operator, and classes, which demonstrates a good understanding of the topics. They correctly indicate that ES6 is preferred due to its enhanced readability and convenience. The mention of browser compatibility as a reason to use ES5 is relevant and shows awareness of practical concerns in project decisions. However, while the candidate's response is correct, it lacks some depth and specific examples of situations where choosing between the two might be critical, which would elevate their answer closer to the 'very good' category.

QUESTION :

Question:

How would you use Git to manage version control in a distributed team environment?

DIALOGUE:

AI Interviewer

Thanks for your detailed answer. How would you use Git to manage version control in a distributed team environment?

Candidate

Git right now is the go-to option for version control in distributed teams because Git allows us to store our project on the server and make changes to it locally. We can push those changes to the server and Git allows us to manage those changes in a way that doesn't interfere with other people's changes. I use it as intended, so when making a change I create a new branch with those functionalities or user stories and make changes there to not mess up the main branch. After those changes are pushed to the server, a pull request can be created, and it can be reviewed by others. They can make comments and ask for changes before pushing the code to the main branch. This way we don't mess up the working code. We can make changes on another branch and push changes to the main code only after we've successfully tested the new functionality. The main benefit is we can make a lot of branches and work in parallel on different functionalities. Team members can work on different functionalities without interfering with each other's changes.

Score:

82

Explanation:

The candidate provides a clear and structured response regarding how to use Git in a distributed team environment. They accurately describe the process of creating branches for new features, making changes, and using pull requests for code reviews, which demonstrates a solid understanding of Git's intended use. Their mention of testing before merging changes into the main branch indicates a good grasp of best practices in version control. However, while their answer is correct, it could benefit from deeper insights into specific challenges or advantages of using Git in a distributed setting, which would elevate it to a very good score.

Conclusion:

Overall, Oleksandr is a solid mid-level candidate with relevant experience and a good understanding of core frontend technologies. The candidate's skills align well with the job requirements, though they may need additional support or ramp-up time for fintech-related projects.

Strengths:

The candidate demonstrates strong competence in frontend development with experience in React and Next.js. They have a good grasp of essential technologies like JavaScript (ES5, ES6), CSS preprocessors, and RESTful APIs. The candidate effectively discusses performance optimization techniques in React, which is crucial for the role.

Weaknesses:

The candidate's explanations, while mostly correct, sometimes lack depth or specific examples, particularly in illustrating how concepts are applied to real-world situations. Additionally, there was a lack of fintech-specific experience and limited discussion on ES5 usage contexts.