Part 2: Frontend Test

1. JavaScript Deep Dive:

Promises in JavaScript represent a potential value that might be available now, in the future, or never. They provide a way to handle asynchronous operations.
A Promise has three states:

- **Pending**: Initial state.
- **Fulfilled**: Completed successfully.
- **Rejected**: Failed.

Here's a basic Promise structure:

```
const ourPromise = new Promise((resolve, reject) => {
    // some asynchronous operation
    if (/* operation successful */) {
        resolve('Success!');
    } else {
        reject('Failure!');
    }
});
```

To handle multiple asynchronous operations sequentially, we chain them using .then()

```
function firstOperation() {
    return new Promise(resolve => {
        setTimeout(() => {
            console.log('First completed');
            resolve();
        }, 1000);
    });
}

function secondOperation() {
    return new Promise(resolve => {
        setTimeout(() => {
            console.log('Second completed');
            resolve();
        }, 1000);
    });
}

firstOperation()
    .then(secondOperation)
    .then(() => console.log('All done sequentially'))
    .catch(error => console.error(error));
```

2. Frontend Performance:

The Critical Rendering Path is the sequence of steps browsers follow to convert HTML, CSS, and JavaScript into visual content on the screen. It involves:

- **DOM Construction**: Parsing HTML to form the Document Object Model (DOM).
- **CSSOM Construction**: Parsing CSS to form the CSS Object Model (CSSOM).
- **Render Tree Construction**: Merging DOM and CSSOM to form the render tree.
- **Layout**: Calculating the position and size of each element.

- **Painting**: Filling in pixels for each element.

Optimization Strategies:

- **Minimize Bytes**: Compress and minify HTML, CSS, and JavaScript files.
- **Reduce Critical Requests**: Inline critical CSS and defer non-critical CSS/JS.
- **Optimize Images**: Use modern formats (e.g., WebP) and serve responsive images.

3. Frontend Code: Please check the code included in the repository