Ruby

Basic Guidelines:

For the problem below, please use Ruby to write a simple dice-rolling program. The code should be unit tested, preferably with RSpec. A CLI or web-based UI is an optional bonus.

The code should demonstrate understanding of clean coding, Ruby, object-oriented and/or functional concepts, and unit testing. The solution will be discussed in a follow-up interview, so be prepared to answer questions about design choices.

Please do not spend more than 1 hour on this problem. The discussion will include the "completeness" of the solution, but that is far less important than the overall code review.

Problem Description:

We would like to have a dice-rolling simulator capable of simulating the rolling of a set of dice of any combination of sides. Some examples:

- Rolling a single 6-sided die.
- Rolling two 8-sided dice.
- Rolling a single 6-sided die and a 20-sided die.

Acceptance Criteria:
Consider the below as the detailed requirements. In most cases, these can be directly applied as unit testing ideas, though they may not cover the complete set of unit tests that should be written.

1. The end product of a "roll" is a result object with a description of what dice were rolled,
2. the value from each die, and the total of all the values.
3. The result should be random. (How to test this is up to the developer.)
4. Rolling a die with n sides should result in a value between 1 and n (inclusive).
5. If a CLI or web UI is added:
    a. The user should be able to specify how many dice are to be rolled.
    b. The user should be able to specify how many sides each die has.
    c. The result should display the separate value for each die as well as the overall total of all values.

Web Design

You have been provided with a design template for a modern e-commerce landing page. The design includes sections such as a hero banner, product showcases, customer testimonials, and a footer. Your task is to convert this design into a fully responsive webpage using **TailwindCSS**.

You only need to focus on the first, third, and final sections, including the footer. Please do not spend don't spend more than 4 hours on this task.

Requirements:

1. **Recreate the Design Using TailwindCSS:**
    - Implement the Figma design in HTML and TailwindCSS, ensuring the design is responsive across different screen sizes (mobile, tablet, desktop).

2. **Responsiveness:**
    - Use TailwindCSS's responsive utilities (sm, md, lg, xl) to adjust the layout and styling based on screen size.
    - Implement the hamburger menu for the navigation on mobile devices.

3. **Code Translation:**
    - Accurately translate the design's spacing, typography, colors, and layout as shown in the Figma template using TailwindCSS classes.
    - Pay attention to details such as padding, margins, and alignment to ensure the code mirrors the Figma design.

4. **Performance Considerations:**
    - Discuss how you would optimize the webpage's performance, particularly focusing on responsive images, minimizing CSS size, and ensuring fast load times on mobile devices.

Deliverables:

- A fully responsive HTML file styled with TailwindCSS that accurately reflects the given design.
- A brief write-up explaining how you approached translating the design into code, including any challenges you faced and how you addressed them.

You need to share all your work through your GitHub profile.