1. Assignment Overview

Title: Image Carousel

Objective: By completing this assignment, you will learn how to create an interactive image slider library using Typescript and HTML/CSS. This will enhance your understanding of DOM manipulation, event handling, and animations.

2. Assignment Learnings

Skills Gained:

- DOM manipulation
- Event handling in Typescript
- CSS animations and transitions
- Object-Oriented Programming (OOP) in Typescript (optional)

Technology Stack:

- TypeScript
- HTML/CSS
- Vite with vanilla-ts template

3. Introduction

Context: Image carousels are a common feature in web design, providing an engaging way to showcase multiple images within a limited space. This assignment involves creating a functional image slider library with various features and customization options.

Relevance: This assignment aligns with the fellowship's goals by providing practical experience with Typescript and enhancing your ability to create dynamic and interactive web components.

4. Problem to Solve

Description: Implement an image slider library that allows users to navigate through images using controls like forward and back arrows and indicator dots. The slider should include smooth sliding animations and be configurable in terms of transition and hold times.

Specifications:

- HTML content structure:

```html
<div class="carousel-container">
  <img src="#" alt="Image" />
  <img src="#" alt="Image" />
  <img src="#" alt="Image" />
  ...
</div>
```

or

```html
<div class="carousel-container">
  <div class="carousel-image-wrapper">
    <img src="#" alt="Image" />
    <img src="#" alt="Image" />
    <img src="#" alt="Image" />
    ...
  </div>
```

```
</div>
```

- Functional controls: forward and back arrows, indicator dots
- Sliding animation for transitions
- Boundary behavior: change direction or animate towards the other boundary and continue
- Clickable indicator dots for image transition

Extra Features:

- Configurable transition and hold times
- Automatic sliding with a fixed interval
- Support for multiple instances of the slider on a single page
- Responsiveness

5. Demo

Example Output:

- An image carousel with functional navigation controls, smooth transitions, and configurable settings.

Figma demo from a community file:
https://www.figma.com/community/file/1047907758163889748

6. Advice

Tips and Tricks:

- Start with the basic HTML structure and incrementally add Typescript functionality.
- Use CSS for smooth animations and transitions.

- Ensure the carousel is responsive and works well on different screen sizes.

7. Walkthrough

Step-by-Step Guide:

1. **Setup**:
    - Initialize your project using Vite with the vanilla-ts template.
    - Create the HTML structure for the carousel.
    - Style the carousel using CSS to ensure it's visually appealing and responsive.
2. **Navigation Controls**:
    - Add forward and back arrows and implement their functionality using TypeScript.
    - Add indicator dots and make them clickable to transition to the corresponding image.
3. **Animations**:
    - Implement sliding animations for smooth transitions between images.
    - Handle boundary conditions to either change direction or animate towards the other boundary.
4. **Extra Features**:
    - Add configuration options for transition and hold times.
    - Implement automatic sliding functionality.
    - Ensure support for multiple instances of the slider on a single page.

Clarifying Questions:

- How can you optimize the sliding animations for performance?
- What are the challenges in making the carousel responsive?

8. How to Test

Testing Criteria:

- The carousel should function correctly with all specified controls.
- Transitions should be smooth, and animations should work as expected.
- The carousel should handle boundary conditions appropriately.
- Extra features should be implemented and configurable as specified.
- Multiple instances of the carousel should work independently.
- The carousel should be responsive and work well on different screen sizes.

9. Submission Guidelines

Format:

- Submit the HTML, CSS, and Typescript files with the repo link.
- Submit the hosted Github pages link.

Platform:

- Submit via the assignment submission form: here

Deadline:

- Due date is Jun 11, 2024 10:00 AM GMT+5:45.