Senior Frontend Engineer (XState)

Advanced Frontend Engineering Take-Home Assessment

This assessment is designed to evaluate your proficiency in advanced frontend development, with a focus on state management, scalability, and real-world problem-solving. The task reflects challenges similar to those faced in developing tools like Lottie Creator.

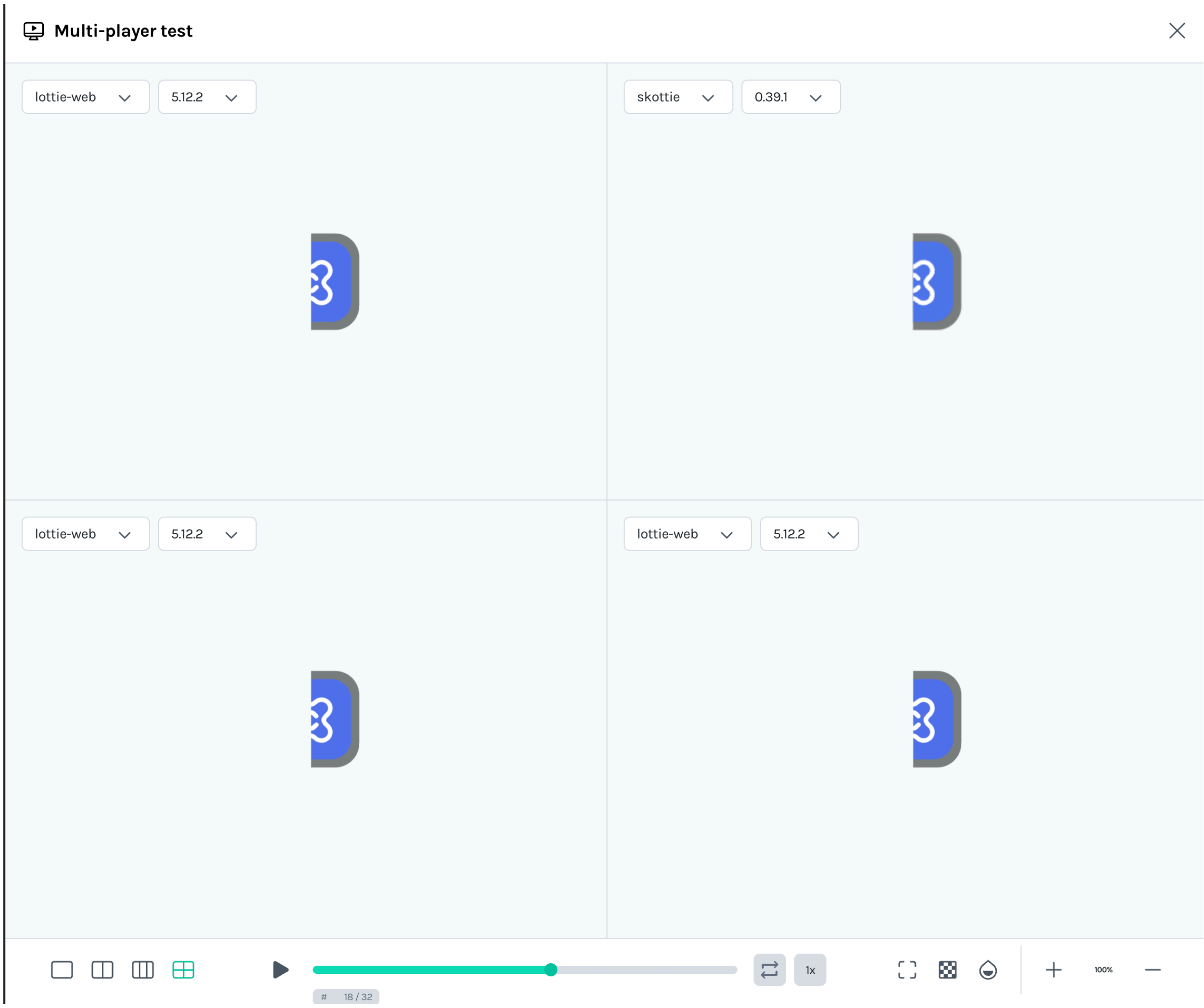
Task Overview

The goal is to create a single-page application that integrates and synchronizes multiple Lottie players and versions, including but not limited to:

1. **DotLottieWeb**
2. Skottie
3. LottieWeb

The application should load a `.lottie` animation file and play it simultaneously across all players. Users should be able to control the animation (play, pause, seek, stop, loop) and apply these controls to all players in a synchronized manner or individually.

Here is a sample UI for inspiration.



Requirements

1. **UI Features:**
   - A control panel to load a `.lottie` animation file.
     - Stretch goal: Find a clean UX to load multiple .lotties and their versions simultaneously
   - Global playback controls (e.g., Play, Pause, Stop, Seek, Zoom) that apply to all players simultaneously.
   - Individual controls for each player.

   - A visual indicator showing the current playback time for each player.
   - Optional: Allow the user to adjust playback speed globally or individually for each player.

2. **State Management:**
   - Use **XState** to manage the state of the application and synchronization across all players.
   - Ensure proper handling of asynchronous events, such as file loading or seeking.
   - Demonstrate how state machines or statecharts are used to model the synchronization of multiple independent players.

3. **Player Integration:**
   - Implement and configure the four specified Lottie players.
   - Ensure seamless integration and consistent behavior across all players.
   - Handle edge cases, such as mismatched player capabilities or unsupported features.

4. **Rendering and Performance:**
   - Demonstrate strategies for optimizing rendering performance with multiple players.
   - Discuss how your solution scales with additional players.

5. **Code Quality and Best Practices:**
   - Use React and TypeScript.
   - Apply modern state management practices.
   - Style the application with Tailwind CSS.
   - Include linting, testing, and adherence to web development best practices.

Stretch Goals

1. **Advanced State Management:**

- Implement detailed transitions and hierarchical states in XState/Stately to handle complex workflows, such as switching between global and individual controls.
- Provide diagrams or visualizations of your state machines.

2. **Performance Optimization:**
   - Optimize the playback experience for all players when rendering a large `.lottie` animation file.
   - Implement lazy loading or efficient resource management techniques to improve performance.
3. **Dynamic File Handling:**
   - Add support for drag-and-drop file loading in addition to a file picker.
   - Validate and handle errors gracefully if the user attempts to load an unsupported file type.
4. **Collaboration Tools:**
   - Implement a system where the state of the players can be exported and imported as a JSON configuration for reproducibility.

Deliverables

- Source code shared via Git (include a `README.md` file with setup instructions).
- Zip file containing the source code.
- Deployment of the solution with an accessible link for evaluation.
- Deliver within 7 calendar days
- Documentation that includes:
  - Diagrams and explanations of the state machine(s).
  - Performance optimization strategies used.
  - How the solution can scale to support more players or additional features.

Evaluation Criteria

- **Technical Proficiency:** Effective use of XState/Stately and understanding of state management principles.
- **Problem Solving:** Ability to integrate and synchronize diverse players.
- **Scalability:** Code organization and architecture to accommodate future features.
- **Code Quality:** Clean, maintainable, and well-documented code.
- **Creativity:** Thoughtful UI design and any additional features that enhance the user experience.