Technical Assignment: Content Platform
Optimized Virtualized Masonry Grid with Detailed Photo View

Objective:

Create a Single Page Application (SPA) that showcases your React skills, particularly in implementing a responsive, optimized, and virtualized masonry grid layout along with a detailed view for photos.

Task Description:

SPA Features:

- **Virtualized Masonry Grid Layout:**
  - Develop a responsive masonry grid that dynamically arranges photos fetched from the Pexels API.
  - Ensure the grid is implemented without using any external libraries for layout management.
  - The grid should be virtualized to efficiently handle large sets of images, improving performance by only rendering visible photos.

  Source: Refer to the Pexels API documentation for guidance on API usage.
- **Photo Details View:**
  - Create a detailed view that displays a selected photo in a larger size, along with additional information such as the title, description, photographer's name, and date taken.
  - Include a back button to return to the masonry grid.

TypeScript:

- Ensure the entire application is written using TypeScript.
- Utilize strong typing, including interfaces and types wherever necessary.
- Implement utility and generic types where appropriate.

React Features:

- Use React hooks effectively (e.g., `useState`, `useEffect`, `useMemo`).
- Demonstrate side effects handling with `useEffect`.
- Use React Router for navigation between the grid and detailed view pages.

Performance:

- The main criteria for evaluation will be performance, with a focus on web vitals metrics, bundle size, unused chunk sizes, and JavaScript execution.

- Use `useMemo` or `useCallback` where necessary to prevent unnecessary re-renders or recalculations.
- Optimize images or any assets you use.

Extras:

- Use styled-components or any CSS-in-JS solution of your choice.
- Ensure the application is responsive.
- Implement error boundaries in the application for better error handling.

Documentation:

- Include a `README.md` detailing how to run and build the project, and any design decisions you made.
- Discuss how you ensured the application's performance and any tools or techniques you used.

Evaluation Criteria:

- **Code Quality:** Consistent code style, clear namings, modularity, and usage of best practices.
- **Functionality:** All features should work without errors.
- **Performance:** Efficient rendering, optimized asset loading, and smooth user experience, with a strong focus on web vitals metrics, bundle size, and JavaScript execution.
- **Responsiveness:** The application should be fully responsive across devices.

Bonus:

- **Test :** Includes unit tests and content search functionality as bonuses.
- **Search Functionality:** Implement a search feature that allows users to search for photos by keywords. The search results should update the masonry grid dynamically, fetching and displaying relevant photos from the Unsplash API.

Submission:

Submit your application as a Git repository (e.g., GitHub, GitLab, Bitbucket). Ensure to include all source files and documentation. Make sure to commit regularly so we can track your development process.