React Developer Intern Test

The goal of this assessment is to assess your ability to build a functional application using React and TypeScript, your proficiency in using data structures and algorithms to efficiently manage and manipulate data, and your ability to use component libraries and CSS frameworks to build a polished and user-friendly UI.

Please read the entire document carefully before starting the assessment.

Project Definition

Build a CRUD (Create, Read, Update, Delete) application using React that allows users to manage a list of tasks. The tasks should have a title, description, due date, priority, and status.

The application should have the following features:

1. A form to add new tasks to the list.
2. A list view that displays all the tasks in the list, sorted by due date and priority.
3. The ability to edit or delete a specific task from the list view.
4. The ability to filter or search for specific tasks in the list view.
5. A visual indicator of task status, such as completed or in progress.

To implement these features efficiently, you will need to use appropriate data structures and algorithms. For example, you could use a priority queue to sort tasks by due date and priority, and a binary search tree to efficiently search for specific tasks.

Using a React component library like Ant, Atlaskit, Fluent UI, or similar will yield extra credits. You can use the components provided by these libraries to build a more polished and user-friendly UI.

Additionally, using Tailwind CSS to style the app will yield extra credits. Tailwind CSS is a utility-first CSS framework that makes it easy to design responsive and modern UIs.

Finally, using TypeScript to build the application will also yield extra credits. You should also consider using Redux or similar state management libraries to handle the application's state and ensure data consistency across different views.

Once you have completed the project, please do the following:

1. Share the link to the git repository where you have stored your code. Make sure the repository is public and accessible to us.
2. Share the link to the hosted application. You can use Vercel, Netlify, or something similar to host the app.
3. Share a walk-through video of the app. You can use Loom, Vimeo, or something similar to record the video. In the video, please demonstrate the functionality of the application and explain the code you have written.