**Round – 1 (Testing analytical and programmatical skills through test)**

**Web Development Challenge – React + JavaScript**

**Position: Junior Web Developer  
Deadline: 15/12/2024**

## Objective

The objective of this challenge is to assess your ability to build clean, functional, and responsive React applications using JavaScript. The project will also test your ability to integrate libraries and implement programmatic solutions.

## Project Description: Task List Manager with Editable Table

You are required to build a Task List Manager with the following features:  
1. Display a Task Table using an editable table library such as Tabulator.js or similar.  
2. Allow users to add, edit, delete, and filter tasks dynamically.  
3. Integrate dummy data using the provided placeholder API.

## Core Requirements

1. **Editable Task Table**  
- Use Tabulator.js to display a table with the following columns:  
 - Task ID  
 - Title  
 - Description  
 - Status: Dropdown with options 'To Do,' 'In Progress,' and 'Done.'  
- Allow inline editing for the Title, Description, and Status fields.  
- Implement a button to delete a task from the table.  
  
2. **Add a New Task**  
- Provide a form or button that allows users to add a new task dynamically.  
- The newly added task should appear in the table without refreshing the page.  
  
3. **Task Filtering**  
- Add a dropdown filter to display tasks based on their Status: 'To Do,' 'In Progress,' or 'Done.'  
  
4. **Fetch Data from Dummy API**  
- Fetch initial task data from:  
 <https://jsonplaceholder.typicode.com/todos>  
- Only display the first 20 tasks.  
- Map 'completed: true/false' to Status ('Done' or 'To Do').  
  
5. Styling  
- Use CSS (or a library like TailwindCSS/Bootstrap) to ensure the table is clean, responsive, and visually appealing.

## Bonus Features (Optional)

1. Add a search bar to filter tasks by their Title or Description.  
2. Add task counters to show the total number of tasks for each status.  
3. Use a toast notification to display success messages when a task is added, edited, or deleted.

## Deployment Requirements

1. Deploy your project to Vercel or Netlify.  
2. Submit:  
 - Live Deployment Link  
 - GitHub Repository Link

## Final Questions

Answer the following questions in a text file and include it in your submission:  
  
1. How did you handle inline editing in the table?  
2. Explain how you fetched and processed the data from the dummy API.  
3. What approach did you use to filter tasks based on status?  
4. How did you manage the state of tasks when adding or editing them?  
5. What challenges did you face during development, and how did you overcome them?

## Submission Guidelines

1. Deploy the project to Vercel/Netlify.  
2. Share the following:  
 - Live Deployment Link  
 - GitHub Repository Link  
3. Include answers to the final questions in a text file.  
4. Email your submission link and answers to [hr@flowlaunch.in] with the subject:  
 **"Frontend Developer Test Submission - [Your Name]"**

## Evaluation Criteria

Your project will be assessed on:  
1. UI/UX Design: Clean, responsive, and visually appealing interface.  
2. Functionality: Task table with working add, edit, delete, and filter features.  
3. Code Quality: Readable, modular, and maintainable code.  
4. Performance: Smooth rendering and responsiveness.  
5. Analytical Skills: Accuracy of answers in the final questions.

## Notes

Ensure the project is functional and deployed before submission.  
Use only the provided API for data fetching.  
Submit within the specified deadline.

Good luck! We look forward to seeing your work. 🚀