**"Frontend Developer Test Submission - [Meghraj Dewangan]"**

**Deployment Link -**  <https://task-list-manager1-34skfngef-meghraj-dewangans-projects.vercel.app>

**GitHub Repository Link -** [**https://github.com/meghraj-dewangan/Task-list-manager1.git**](https://github.com/meghraj-dewangan/Task-list-manager1.git)

**1. How did you handle inline editing in the table?**

- I used ReactTabulator for inline editing by enabling the editor property for columns like taskid, title, description, and status. Changes are detected the cellEdited event, which updates the tasks state with the new values

**2. Explain how you fetched and processed the data from the dummy API?**

- I fetched data from the dummy API using the fetch function and retrieved the first 20 tasks. Each task was processed to include id, title, and status (Done or To Do based on completion).

**3. What approach did you use to filter tasks based on status?**

**-** To filter tasks by status, I used the filter method on the tasks array. If a status filter is set, only tasks that match the selected status are shown. If no filter is applied, all tasks are displayed. The filtered tasks are then saved in the filteredTasks variable and shown in the table.

**4. How did you manage the state of tasks when adding or editing them?**

* I used the useState hook to manage tasks. For adding, I created a new task object and updated the tasks array using the spread operator. For editing, I used the cellEdited event to detect changes, updated the specific task in the array, and saved the updated list with setTasks.

**5. What challenges did you face during development, and how did you overcome them?**

- I faced challenges with handling API errors, updating tasks correctly, and filtering data. To fix these, I used try-catch for error handling and showed error messages with react-toastify. I also made sure only the edited tasks were updated by mapping through the list. For debugging, I used console.log to track issues.