ProU Technology

Frontend - Web Development Assignment

Internship Coding Challenge – Track 1: Frontend (Mock Data)

1. Scenario

You've joined a small startup that's building an **Employee Task Tracker** web app.
The backend is still under development, but your task is to **design and build a functional frontend** using **mock JSON data**.

The app will allow users to:

- View a list of employees and their assigned tasks
- Filter tasks by status (Pending / In Progress / Completed)
- Add new tasks (frontend-only)
- View a dashboard summary (e.g., total tasks, completed %, etc.)

You can use **any frontend framework** you are comfortable with (React / Vue / Angular).
Focus on clean design, good code structure, and interactivity.

2. Expected Outcome

A **single-page responsive web application** that:

- Displays employees and their tasks
- Allows filtering and basic task addition (mock data only)
- Shows a dashboard summary
- Uses a clean, modern UI (e.g., Bootstrap, Tailwind, or custom CSS)

3. Deliverables

Please submit:

1. A **GitHub repository** with the source code
2. A **README.md** that includes:
   - Project setup instructions
   - Framework and libraries used
   - Any assumptions made
3. A short **screen recording (max 2 minutes)** demonstrating the app

4. Instructions

- You may use **mock data** stored locally in a .json file or hardcoded in code.
- No backend or database connection is required.
- Use modern best practices:
  - Component-based structure
  - State management (e.g., React useState/useEffect or Vue reactive data)
  - Clean, consistent naming and indentation
- Write readable and maintainable code.

5. Focus Areas (Evaluation Criteria)

Area	What we look for
UI Design	Modern layout, color balance, and responsive design
Code Quality	Clean structure, comments, reusability, and best practices
Data Handling	Proper use of mock data, filtering, and conditional rendering
Interactivity	Buttons, forms, and visual feedback that feel natural
Creativity	Small touches that improve UX (e.g., animations, icons, hover effects)

6. Bonus Challenge (Optional)

Implement **local storage persistence** so that when the user adds a new task or changes status, the data persists after a page refresh.

7. Estimated Time

4–6 hours total (you can complete it in 1–2 sittings).

8. Example Mock Data

```
{
     "employees": [
      {
        "id": 1,
        "name": "Alice Johnson",
        "role": "Frontend Developer",
        "tasks": [
          { "id": 101, "title": "Build login page", "status": "Completed" },
          { "id": 102, "title": "Implement dashboard", "status": "In Progress" }
        ]
      },
      {
        "id": 2,
        "name": "Bob Smith",
        "role": "Backend Developer",
        "tasks": [
          { "id": 103, "title": "API integration", "status": "Pending" }
        ]
      }
     ]
}
```