

# Frontend Assignment

## Dashboard with Role-Based Views (Redux Toolkit Implementation)

Stack: React (with Hooks), Redux Toolkit, Tailwind CSS or plain CSS, Optional: Recharts/Chart.js

### Project Overview:

**Build a Team Pulse Dashboard**—a productivity monitoring tool for an internal team. This dashboard allows Team Leads to monitor member statuses and assign tasks, while Team Members can update their status and manage progress on tasks. You must use React for UI rendering and Redux Toolkit for managing all application state globally. No backend integration is required, but you may use a free, open-source API (such as <https://randomuser.me/>) to simulate initial team member data.

### Requirements:

📌 Requirements Summary:	
Feature	Description
Role Switching	Toggle between <code>Team Lead</code> and <code>Team Member</code> views
Status Update	Members can update their live working status
Task Management	Leads assign tasks, members update progress
Redux Toolkit	Use <code>createSlice</code> , <code>configureStore</code> , <code>useSelector</code> , <code>useDispatch</code>
UI Library	Tailwind CSS or your own custom CSS
State Persistence (Optional)	Redux Persist or <code>localStorage</code>
Bonus (Optional)	Auto-reset, Chart visualization, Dark mode

### User Roles & Functionalities:

#### Role Switching:

A toggle (top-right corner) should switch between Team Lead and Team Member mode. The current role and user must be tracked in the Redux state.

## **Team Lead View:**

### 1. Team Member Status Monitor:

- List all members with their current status badge: Working, Break, Meeting, Offline.
- Summary section (e.g., “2 Working · 1 Meeting · 1 Break”).

### 2. Assign Tasks to Members:

- Task form with: Member selection dropdown, Task title input, Due date picker
- On submission, update Redux state to assign task to selected member.

### 3. Filtering & Sorting:

- Filter team members by status.
- Sort members by number of active (not completed) tasks.

## **Team Member View:**

### 1. Update Your Status:

- Buttons to select: Working, Break, Meeting, Offline
- Only one can be active; update state globally

### 2. View Your Tasks:

- Show a list of assigned tasks.
- Each task should display title and due date, include a progress bar (0–100%), allow increment/decrement in steps of 10%, and be marked completed once progress reaches 100%

## **Redux Toolkit Usage (Mandatory):**

Use Redux Toolkit for all state management. At minimum:

Store Setup:

```
// src/redux/store.js
import { configureStore } from '@reduxjs/toolkit';
import membersReducer from './slices/membersSlice';
import roleReducer from './slices/roleSlice';

export const store = configureStore({
  reducer: {
    members: membersReducer,
```

```
        role: roleReducer
    }
});
```

Example Slice Structure:

```
// src/redux/slices/roleSlice.js
import { createSlice } from '@reduxjs/toolkit';

const roleSlice = createSlice({
    name: 'role',
    initialState: {
        currentRole: 'member',
        currentUser: 'John Doe'
    },
    reducers: {
        switchRole: (state, action) => {
            state.currentRole = action.payload;
        },
        setUser: (state, action) => {
            state.currentUser = action.payload;
        }
    }
});

export const { switchRole, setUser } = roleSlice.actions;
export default roleSlice.reducer;
```

## Suggested Project Structure:

```
/src
  /components
    - Header.jsx
    - MemberCard.jsx
    - TaskForm.jsx
    - TaskList.jsx
    - StatusSelector.jsx
  /redux
    /slices
      - membersSlice.js
      - roleSlice.js
```

```
- store.js  
/pages  
  - Dashboard.jsx  
App.jsx  
index.js
```

## Deliverables:

- GitHub repository with clean commit history
- Live deployment (Netlify, Vercel, etc.)
- README.md with: Project Overview, Tech Stack, Setup instructions, Screenshots or demo video (optional)

## Bonus Challenges (Optional):

- Auto Reset: Reset a user's status to Offline after 10 minutes of inactivity.
- Chart View: Visualize status distribution with a pie chart using Chart.js or Recharts.
- Dark Mode: Add a toggle to switch between light and dark themes using Tailwind.

## Design Inspiration – Dashboard Layout Reference

The above UI represents a clean and structured admin dashboard layout that you may refer to while designing the **Team Pulse Dashboard**. Key layout ideas you can take from this design include:

- **Left Sidebar Navigation:** Use a sidebar for switching between different dashboard sections (if needed).
- **Header Section:** Displays user info (like name/role) and optional toggles (dark mode, settings).
- **Card-Based Metrics:** Use cards to show quick summaries such as employee status counts, progress summaries, or pulse distributions.
- **Visualizations:** Integrate simple graphs (like line charts or pie charts) for visual clarity.

- **Component Consistency:** Each module (e.g., "Upcoming Interviews" or "Total Employees") is separated into its own clean box, promoting modularity.

While you don't need to replicate this design exactly, you are encouraged to adopt its **structure, spacing, visual hierarchy, and clarity** when building your own dashboard.

