# Technical Assignment

## Request & Approval Workflow System

**React + Node.js**

---

### Overview

This assignment is designed to evaluate how you think about **workflow logic, state management, and clean system design**.

You'll build a small system where users can submit requests and other users can review, approve, or reject them.
The goal is **not** to build a large or complex product, but to demonstrate **clear reasoning, correctness, and good structure**.

---

## What You'll Build

A simple **Request & Approval Workflow System** with:

- A React frontend
- A Node.js backend
- Basic persistence (database)

---

## Core Concepts

### Users

There are two roles:

- **Requester**
- **Approver**

A user can have **one or both roles**.

---

### Requests

Each request should contain:

- **Title** (required)
- **Description** (optional)
- **Type** (e.g. Access, Finance, General)
- **Status**
  - Draft
  - Submitted
  - Approved
  - Rejected
- **Created by**
- **Created / Updated timestamps**

---

# Functional Requirements

### As a Requester, I should be able to:

- Create a new request
- Edit or delete a request **only while it is Draft**
- Submit a request
- View a list of my own requests
-

### As an Approver, I should be able to:

- View all **submitted** requests
- Approve a request
- Reject a request
- Add a comment when approving or rejecting

# Business Rules (Very Important)

These rules **must be enforced**:

- Only **Draft** requests can be edited or deleted
- Once a request is **Submitted**, it cannot be modified by the requester
- Only users with the **Approver** role can approve or reject requests
- **Rejected** requests cannot be re-submitted
- All request **status transitions must be validated server-side**

# Frontend Requirements (React)

Your UI should clearly communicate the workflow and user roles.

### Navigation

Include navigation between:

- **My Requests**
- **Pending Approvals**

### Request List

Each request item should show:

- Status
- Type
- Created date

### Request Details View

- Full request information
- Approve / Reject actions (visible only to Approvers)
- Display of approver comments (if any)

### UX Expectations

- Clear handling of:
  - Loading states
  - Empty states
  - Error states
- Confirmation before approving or rejecting a request

---

# Backend Requirements (Node.js)

Your backend should focus on correctness and clarity.

It must handle:

- Validation of request status transitions
- Role-based access control
- Clear, consistent error responses
- Data persistence: use a database

---

# Testing (Light, but Required)

You are **not** expected to write extensive tests.

Minimum expectations:

- Tests for request status transitions
- Tests preventing invalid actions (e.g. editing after submission)
- Tests for approval and rejection rules

---

# Constraints

- **Timebox:** Up to **4 days**
- Please keep the scope focused:
  - No notifications
  - No external integrations

---

# Deliverables

Please submit:

- A Git repository containing:
  - Frontend (React)
  - Backend (Node.js)
- Seed data including:
  - A few users with different roles
  - Sample requests in different states
- A **README** that includes:
  - Setup instructions
  - Key decisions and tradeoffs
  - Assumptions you made
  - What you would improve with more time

---

# Nice to Have (Optional)

These items are **not required** and will not negatively impact your evaluation if omitted.

- **Deployment**
  - Deploy the frontend and/or backend to any platform of your choice (e.g. Vercel, Netlify, Render, Railway, Fly.io, etc.)
  - Include the deployed URL(s) in the README
  - Basic environment configuration is sufficient

# How We Evaluate

We'll be looking at:

- Correctness of workflow logic
- Backend validation and error handling
- Frontend clarity and UX state handling
- Code readability and structure
- Separation of concerns

---

### Final Notes

- You are free to make reasonable assumptions — just document them.
- Simplicity and correctness are valued over extra features.
- We care more about **how you think** than how much you build.

Good luck — we're looking forward to reviewing your solution.