Workflow Builder: Project Approach

Project Overview

I built a workflow builder application for quality management systems. It allows users to create approval workflows with customizable steps, conditions, and role assignments. The application helps quality managers visualize and organize their approval processes.

Database Design

I implemented a MySQL database with four main tables: workflows, workflow_steps, step_connections, and step_roles. Each table has appropriate relationships to maintain data integrity while providing flexibility for different workflow structures.

Backend Development

The Node.js/Express backend uses a modular architecture with separate models and controllers. The API follows RESTful principles with endpoints for managing workflows, steps, connections, and role assignments. Database operations are isolated in model files for better maintainability.

Frontend Implementation

I created a responsive user interface using Next.js and Tailwind CSS. The application includes pages for listing workflows and designing individual workflows. Components are organized by function (layouts, workflows, steps) for better code organization.

Key Features Implemented

- Workflow creation and management with active/inactive status
- Step builder with different types (approval, task, notification)
- Conditional logic between steps (always, if approved, if rejected)
- Role assignment to track responsibility for each step
- Workflow visualization showing steps and connections
- Export functionality for sharing workflows

Challenges Faced

The main challenges included creating an intuitive UI for building workflows, managing complex data relationships, and ensuring proper integration between frontend and backend. I had some difficulty with role assignment functionality but implemented workarounds to demonstrate the core concepts.

Design Decisions

I chose a card-based visualization approach for simplicity and better user experience. I implemented a straightforward condition model with three options (always, if approved, if rejected) to keep the interface intuitive while providing enough flexibility for typical approval workflows.