Enterprise Workflow Dashboard - Comprehensive Project Design

Document 1: Enterprise Workflow Dashboard - Comprehensive Project Design

Project Title:
Enterprise Workflow Dashboard

1. Project Overview

Purpose:
An Angular-based enterprise-grade application designed to manage and monitor workflows across departments. It provides real-time task tracking, KPI dashboards, and role-based access for users such as Admins, Managers, and Employees.

Key Features:
- Task/workflow creation, assignment, and tracking
- Role-based dashboards with analytics
- Real-time updates via WebSockets simulation
- Approval chains and customizable steps
- Search, filter, and pagination
- Notification system
- JWT-based authentication
- Audit logs and activity history

2. Data Structure

Models:

```typescript
export interface User {
  id: string;
  name: string;
  email: string;
  role: 'Admin' | 'Manager' | 'Employee';
  department: string;
}
export interface Task {
  id: string;
  title: string;
  description: string;
  assignedTo: string;
  createdBy: string;
  status: 'Pending' | 'In Progress' | 'Completed' | 'Rejected';
  priority: 'Low' | 'Medium' | 'High';
  dueDate: Date;
  createdAt: Date;
  updatedAt: Date;
}

export interface Workflow {
  id: string;
  name: string;
  steps: WorkflowStep[];
  currentStep: number;
  createdBy: string;
  createdAt: Date;
}

export interface WorkflowStep {
  id: string;
  title: string;
  assignedToRole: string;
  status: 'Pending' | 'Approved' | 'Rejected';
  notes?: string;
}

export interface Notification {
  id: string;
  userId: string;
  message: string;
  read: boolean;
  createdAt: Date;
}
```

API Endpoints (Simulated via in-memory API):
- `/api/users`
- `/api/login`
- `/api/tasks`
- `/api/workflows`
- `/api/notifications`
- `/api/audit-log`

3. Screens & UI Components

Screens:
1. Login
2. Dashboard
3. Tasks (List, Details, Form)
4. Workflow Builder
5. User Management
6. Notifications
7. Audit Logs

Component Examples:
- task-list, task-item, task-form
- workflow-builder, workflow-step
- chart-kpi, notification-bell
- user-table, user-form
- auth-login, header, sidebar

4. Angular Concepts Used

- Components, Modules, Routing, Guards
- Services, HttpClient, Observables
- Reactive Forms, Template-driven forms
- Lazy Loading, Route Guards, Interceptors
- Angular Material, Custom Pipes, Directives
- Lifecycle Hooks, Animations
- State Management with RxJS (BehaviorSubject)

5. User Roles

Roles & Permissions:
- **Admin**: Full access, manage users, view logs
- **Manager**: Manage team workflows/tasks
- **Employee**: View and update personal tasks

6. Best Practices & Scalability

- Modular architecture with feature modules
- Shared and Core modules for reusability
- Strict typing and Angular best practices
- Dependency injection and service abstraction
- Barrel files and environment configs
- Production build with lazy loading and AOT

7. Additional Features

- JWT-based Auth using Interceptors
- Real-time UI simulation with RxJS
- Export data to CSV/PDF
- Dark/light mode toggle
- Comprehensive audit trail

8. Project Structure

```
src/
├── app/
│   ├── core/            Auth, guards, interceptors
│   ├── shared/            Pipes, directives, components
│   ├── features/
│   │   ├── auth/
│   │   ├── dashboard/
│   │   ├── tasks/
│   │   ├── workflows/
│   │   ├── users/
│   │   ├── notifications/
│   ├── app-routing.module.ts
│   ├── app.module.ts
```

Deployment:

- Angular build optimization (`ng build --prod`)
- Deploy via Netlify, Vercel, or Firebase
- GitHub repo with full documentation

Final Outcome:

- **Fully functional enterprise-style Angular SPA**
- **Strong demo for Senior Angular Developer portfolio**
- **Modular, testable, and scalable codebase**