**��️ CourseWorx Tech Stack & System Architecture**

**📋 Complete Technology Stack**

**🎯 Frontend Technologies**

* **React 18.2.0** - Core UI framework
* **React Router DOM 6.20.1** - Client-side routing
* **React Query 3.39.3** - Server state management & caching
* **React Hook Form 7.48.2** - Form handling and validation
* **React Beautiful DnD 13.1.1** - Drag and drop functionality
* **React Hot Toast 2.4.1** - Toast notifications
* **React i18next 15.6.1** - Internationalization
* **React Quill 2.0.0** - Rich text editor
* **React Video JS Player 1.1.1** - Video player component

**🎨 UI & Styling**

* **Tailwind CSS 3.3.6** - Utility-first CSS framework
* **Headless UI 1.7.17** - Unstyled, accessible UI components
* **Heroicons 2.0.18** - Icon library
* **Video.js 8.23.4** - Video player with themes
* **@videojs/themes 1.0.1** - Video.js themes

**🔧 Development Tools**

* **Create React App (CRA)** - React application boilerplate
* **CRACO 7.1.0** - Create React App Configuration Override
* **PostCSS 8.4.32** - CSS post-processor
* **Autoprefixer 10.4.16** - CSS vendor prefixing
* **ESLint** - Code linting and quality
* **Jest** - Testing framework
* **Testing Library** - React testing utilities

**🌐 Backend Technologies**

* **Node.js** - JavaScript runtime
* **Express.js 4.18.2** - Web application framework
* **Sequelize 6.35.1** - SQL ORM
* **PostgreSQL** - Primary database
* **SQLite** - Development database (fallback)

**🔐 Security & Authentication**

* **JSON Web Tokens (JWT) 9.0.2** - Authentication tokens
* **bcryptjs 2.4.3** - Password hashing
* **Helmet 7.1.0** - Security headers
* **CORS 2.8.5** - Cross-origin resource sharing
* **Express Validator 7.0.1** - Input validation

**�� File Management**

* **Multer 1.4.5** - File upload handling
* **FFprobe Static 3.1.0** - Video metadata extraction
* **CSV Parser 3.2.0** - CSV file processing

**🛠️ Development & Build Tools**

* **Nodemon 3.0.2** - Development server auto-restart
* **Concurrently 8.2.2** - Run multiple commands simultaneously
* **Axios 1.6.2** - HTTP client
* **Moment.js 2.29.4** - Date manipulation
* **Dotenv 16.3.1** - Environment variable management

**🗄️ Database**

* **PostgreSQL 8.11.3** - Primary production database
* **pg-hstore 2.3.4** - PostgreSQL hstore support
* **SQLite** - Development and fallback database

**🏛️ System Architecture Overview**

text

┌─────────────────────────────────────────────────────────────────┐

│                        COURSEWORX SYSTEM                        │

└─────────────────────────────────────────────────────────────────┘

┌─────────────────┐    ┌─────────────────┐    ┌─────────────────┐

│   FRONTEND      │    │    BACKEND      │    │   DATABASE      │

│   (Port 3000)   │◄──►│   (Port 5000)   │◄──►│   (PostgreSQL)  │

└─────────────────┘    └─────────────────┘    └─────────────────┘

         │                       │                       │

         │                       │                       │

    ┌────▼────┐              ┌───▼────┐               ┌────▼────┐

    │  React  │              │ Node.js│              │  SQLite │

    │   App   │              │Express │              │(Dev DB) │

    └─────────┘              └────────┘              └─────────┘

**🔧 Detailed Architecture Components**

**Frontend Architecture (React SPA)**

**text**

┌─────────────────────────────────────────────────────────────┐

│                    FRONTEND LAYER                           │

├─────────────────────────────────────────────────────────────┤

│  Presentation Layer (React Components)                     │

│  ├── Pages (CourseCreate, CourseDetail, Dashboard, etc.)   │

│  ├── Components (CourseSidebar, VideoPlayer, Modals)       │

│  ├── Hooks (useContentManagement, useFileUpload)           │

│  └── Contexts (AuthContext)                                │

├─────────────────────────────────────────────────────────────┤

│  State Management                                           │

│  ├── React Query (Server State)                            │

│  ├── React Hook Form (Form State)                          │

│  └── React Context (Global State)                          │

├─────────────────────────────────────────────────────────────┤

│  Styling & UI                                               │

│  ├── Tailwind CSS (Utility Classes)                        │

│  ├── Headless UI (Accessible Components)                   │

│  └── Heroicons (Icon Library)                              │

├─────────────────────────────────────────────────────────────┤

│  Routing & Navigation                                       │

│  └── React Router DOM (Client-side Routing)                │

└─────────────────────────────────────────────────────────────┘

**Backend Architecture (Node.js/Express)**

**text**

┌─────────────────────────────────────────────────────────────┐

│                    BACKEND LAYER                            │

├─────────────────────────────────────────────────────────────┤

│  API Layer (Express.js)                                    │

│  ├── Routes (auth, courses, content, users, etc.)          │

│  ├── Middleware (auth, validation, CORS, security)         │

│  └── Controllers (Business Logic)                          │

├─────────────────────────────────────────────────────────────┤

│  Data Access Layer                                         │

│  ├── Sequelize ORM                                         │

│  ├── Models (User, Course, Content, Enrollment)            │

│  └── Migrations (Database Schema)                          │

├─────────────────────────────────────────────────────────────┤

│  Security & Authentication                                 │

│  ├── JWT Token Management                                  │

│  ├── bcryptjs Password Hashing                             │

│  ├── Helmet Security Headers                               │

│  └── CORS Configuration                                    │

├─────────────────────────────────────────────────────────────┤

│  File Management                                           │

│  ├── Multer File Upload                                    │

│  ├── Static File Serving                                   │

│  └── Media Processing (FFprobe)                            │

└─────────────────────────────────────────────────────────────┘

**Database Schema (PostgreSQL)**

**text**

┌─────────────────────────────────────────────────────────────┐

│                   DATABASE LAYER                            │

├─────────────────────────────────────────────────────────────┤

│  Core Entities                                             │

│  ├── Users (Trainers, Trainees, Admins)                    │

│  ├── Courses (Online, Classroom, Hybrid)                   │

│  ├── CourseContent (Videos, Articles, Quizzes, etc.)       │

│  ├── CourseSections (Content Organization)                 │

│  └── Enrollments (User-Course Relationships)               │

├─────────────────────────────────────────────────────────────┤

│  Supporting Tables                                         │

│  ├── CourseStats (Analytics & Metrics)                     │

│  ├── UserNotes (Personal Notes)                            │

│  ├── LessonCompletion (Progress Tracking)                  │

│  ├── QuizQuestions (Assessment Data)                       │

│  └── Attendance (Classroom Tracking)                       │

└─────────────────────────────────────────────────────────────┘

**🔄 Data Flow Architecture**

text

User Interaction → React Frontend → API Gateway → Express Backend → Database

     ↑                                                                    ↓

     └─────────── Response ←─── Business Logic ←─── Data Processing ←────┘

**📁 Project Structure**

text

CourseWorx/

├── frontend/                 # React SPA

│   ├── src/

│   │   ├── components/       # Reusable UI components

│   │   ├── pages/           # Route components

│   │   ├── hooks/           # Custom React hooks

│   │   ├── contexts/        # React contexts

│   │   ├── services/        # API service layer

│   │   └── utils/           # Utility functions

│   ├── public/              # Static assets

│   └── build/               # Production build

├── backend/                 # Node.js API

│   ├── routes/              # API endpoints

│   ├── models/              # Sequelize models

│   ├── middleware/          # Express middleware

│   ├── config/              # Configuration files

│   ├── migrations/          # Database migrations

│   ├── scripts/             # Utility scripts

│   └── uploads/             # File storage

└── docs/                    # Documentation

**�� Key Features & Capabilities**

**Course Management**

* Multi-type courses (Online, Classroom, Hybrid)
* Rich content support (Videos, Articles, Quizzes, Documents)
* Drag-and-drop content organization
* Progress tracking and analytics

**User Management**

* Role-based access control (Trainer, Trainee, Super Admin)
* JWT-based authentication
* User enrollment and assignment
* Personal notes and progress tracking

**Media Handling**

* Secure video streaming with authentication
* File upload and management
* Rich text editing with React Quill
* Professional video player with security features

**Internationalization**

* Multi-language support (English, Arabic, French, etc.)
* RTL layout support for Arabic
* Dynamic language switching

**🔧 Development Workflow**

1. **Development**: Concurrent frontend/backend development
2. **Database**: Sequelize migrations for schema management
3. **Build**: CRACO for custom webpack configuration
4. **Testing**: Jest and Testing Library for unit tests
5. **Deployment**: Production-ready builds with optimization

This architecture provides a scalable, maintainable, and feature-rich learning management system with modern web technologies and best practices.

Review Changes

86.8%

6

1 Tab