Tech Stack Overview

Backend (API & Data Layer)

- .NET 8 Minimal APIs (clean endpoints, middleware pipeline, dependency injection)
- C#
- Entity Framework Core (ORM, LINQ queries, async access, relationship handling, provider abstraction)
- Databases: SQLite (local development), PostgreSQL (production)
- Authentication and Security: JWT-based stateless authentication, claims-based user scoping, ownership enforcement
- Cloud Storage (JobTracker): AWS S3 with presigned URLs; metadata stored in database, files stored in object storage

Frontend (UI & Client Layer)

- React (component-based architecture, predictable data flow, hooks)
- TypeScript (strong typing across frontend and backend DTOs)
- Vite (fast development server and build tooling)
- Tailwind CSS (utility-first styling, responsive layouts)
- Axios (centralized API client with JWT interceptors)
- React Router (client-side routing, URL state restoration)
- Drag and Drop (JobTracker): Kanban-style board interactions

Architecture & Design Patterns

- Client–server architecture
- Separation of concerns (API layer, frontend service layer, UI components)
- RESTful API design
- DTO mapping
- Asynchronous programming using async/await
- Horizontal, feature-based project organization
- Stateless backend design

DevOps & Deployment

- Docker

- Render (cloud hosting)
- Environment-based configuration
- Development and production parity
- Git and GitHub (version control, public repositories, documentation)

Project Context

- Task and Rewards Management System: role-based access (Parent/Kid), task creation, completion, and rewards workflow
- JobTracker: Kanban-based job application workflow, attachments system, secure file uploads, ownership-based data isolation