

# Coding Assignment - Collaborative Todo Application

## Goal

Build a small, production-minded Todo application that supports multiple users and basic collaboration.

You are free to make reasonable assumptions where the specification is not explicit. If something feels underspecified, choose an approach you believe makes sense or discuss it with Olle ([olle@arkivia.se](mailto:olle@arkivia.se)) or Christian ([christian@arkivia.se](mailto:christian@arkivia.se)). They regularly check their emails.

---

## Tech Stack

Please use the following technologies as a base:

- **Frontend:** React + TypeScript
- **Backend:** Express + TypeScript
- **Data & Auth:** Firebase / Firestore

Within those constraints, you are free to decide on libraries, patterns, and project structure.

---

## High-Level Feature Description

Write a simple web application where users can:

1. **Register and log in to the system.**
2. **Create and manage TODO items.**
3. **Collaborate on TODOs by assigning them to other users.**

The app should support multiple users using the system at the same time. You are free to decide further aspects of the application as you deem necessary to make the application valuable for the user.

---

# Functional Requirements

At a minimum, your application should support:

## 1. User Accounts

- Users can **sign up and log in**.
- Once logged in, the user sees and manages TODOs relevant to them.
- You can decide what user profile information is needed (e.g. name, email, etc.).

## 2. TODO Management

- A TODO should at least have:
  - A human-readable identifier (e.g. title or name).
  - Some way to know if it is still active or done (you can define states as you like).
- Users should be able to:
  - **Create** TODOs.
  - **Update** TODOs (e.g. change title, status, etc.).
  - **Delete** TODOs.

You decide what additional fields (description, due date, tags, etc.) are useful.

## 3. Ownership & Visibility

- Each TODO should belong to at least one user (the **creator** or **owner**).
- Users should **only be able to manage**:
  - TODOs they own, and
  - TODOs that have been explicitly **assigned to them** by someone else.
- “Manage” here includes reading the details of the TODO, updating it, and deleting it - but you can refine these rules if you think it makes sense.

The exact rules you choose around who can see or edit what (e.g. can assignees delete, can only owners delete, etc.) are up to you - just document your decision.

## 4. Assignment / Collaboration

- A user should be able to **assign a TODO to one or more other users** so that they can co-work on it.
- The UI should make it possible to pick which user(s) to assign to.
  - How you implement user selection is your choice.

---

## Non-Functional Expectations

- **TypeScript:** Use TypeScript on both frontend and backend.
- **Data & API contracts:** Ensure the frontend and backend agree on the shape of data being sent and received. How you achieve this is up to you.
- **Project structure:** Organize your code in a way that you feel would scale beyond a toy app with a single developer.

We are interested not only in “does it work?” but also “would this be maintainable if the app kept growing?”

---

## UI / UX

You may use any UI library (or make your own).

---

## Deliverables

Please provide:

1. **Source code** in Git repository/repositories (GitHub, GitLab, etc.).
2. A short **README** that includes:
  - How to set up and run the project locally.
  - Any assumptions or design decisions you made (especially where the spec was not explicit).
  - Anything you'd do next if you had more time.

If you want to host a demo somewhere (Firebase Hosting, Vercel, etc.), feel free to include a link, but this is optional.

---