

Objective:

Working with complex forms using libraries.

Deadline: 3 days

Task:**Requirements:**

Create an application for listing and managing entities (the entity is up to you; it can be contacts, cards, projects, company — whatever you want).

- The application should have the following pages: List page and create/edit page.
- Create and edit pages can be two different pages, but components should be reusable as much as possible.
- The list page displays a table with an edit button for each row. There should also be a create button (which should navigate to the create form).
- The server should store entities in memory. It should have some predefined default values for easier testing and editing.

The form should have:

- At least 2 select inputs and 4 text inputs
- Optional fields
- Radio button that controls additional two inputs (1 select, 1 text).
- Array inputs: Ability to add multiple nested entities (for example, for projects, a user can add tasks. Each task should have a “name” input and “status” select).

Form validation. Use the “[yup](#)” library.

Required rules to be implemented:

- At least 1 text input should be required

- At least 1 text input should have max length validation
- At least 1 select input should be required

When a user tries to submit the form with failed validation, each input should have a red inline error text below it.

Form submission:

- Values are sent to the server and stored in memory.
- The request payload should contain only what the user filled/changed.
 - On create: if a user filled 4 fields, the request payload must contain only 4 fields.
 - On edit: if a user changed only one field, the request payload must contain 1 field.

Tech stack:

React, react-hook-form, yup, axios, vite, express