Fullstack code challenge

The task of this code challenge is to add functionality described below to this existing sample application: https://github.com/dfds-frontend/fullstack-dev-challenge. The application is built using NextJS, Typescript, Prisma, React Query Tailwind, zod and https://ui.shadcn.com for ready-made components. It is expected that you will complete the below tasks using the technologies listed.

The challenge consists of a variety of frontend, backend and data modeling tasks. We invite you to tailor your implementation as close to the description as possible, otherwise be sure to document deviations if any.

Task 1 - Create new voyage

At the root of the application, place a button “Create” on the top left of the list of mock voyages. When pressed, the button should open https://ui.shadcn.com/docs/components/sheet with the form for creating a voyage inside The form should have the following validations: All fields are required Departure date and time should be before arrival date and time Refresh the list of voyages once a voyage has been successfully created Display https://ui.shadcn.com/docs/components/toast with a success message.

Task 2 - Introduce UnitType relation to Voayge

Enable adding at least 5 UnitTypes to a voyage; existing database table available. Update the Prisma schema accordingly. Ensure selection of minimum 5 UnitTypes.

Task 3 - Modify the list

Add a "unittypes" column to the voyages table. Display unittype count for each voyage. Clicking on unittype count opens a PopOver showing selected UnitTypes: https://ui.shadcn.com/docs/components/popover

List to include:

- □ Name
- □ Default length

Task 4 - Handling voyage deletion error

You may have noticed that deleting a voyage does not always work. Add error handling to inform the user when that happens. It is sufficient to show https://ui.shadcn.com/docs/components/toast with the appropriate error message.