Axon JS Test Task

Prerequisites

This repository contains server and boilerplate client-side code: https://github.com/vlad-khitev-axon/js-test-task.

To download, install and run application, run:
git clone https://github.com/vlad-khitev-axon/js-test-task.git && cd js-test-task && npm install && npm start

As you will see, there is a file *server.js* in the root of repository containing handlers for 4 CRUD operations on user entity. They are:

- GET /users - get list of all users
- POST /users - create new user
- PUT /users/:id - update existing user
- DELETE /users/:id - delete existing user

There is a file *data.json* which is a simplest database accessed synchronously from that handlers. Backup of this file is located in *backup/data.json.*

Libraries that may be used to accomplish the task are up to you. You may want to use libraries included in *package.json*: axios for doing http requests and moment for time manipulation.

Task

Create a page consisting of three elements:

1. Table on which user data is displayed.
2. Form for creating new user.
3. Block with table summary.

Elements description:

1. Table
    a. Display table of users with the following cells:
        i. First name
        ii. Last name
        iii. Date of birth in format DD/MM/YYYY
        iv. Location
    b. Data should be loaded from server. No pagination, sorting etc. needed.

    c. Each row should have a “delete” button in front of it. On clicking “delete” button user should be deleted from database and the table (with summary) should be refreshed.
2. Form
    a. Form should consist of 3 text inputs, 1 date input for date-of-birth and 1 submit button.
    b. Submit button should be disabled until all fields are filled.
    c. On clicking submit button new user should be created.
    d. After successful submit the table (with summary) should be refreshed.
3. Summary
    a. Display summary in any convenient way. You may want to use <dl> <dt> <dd> for it. Summary consists of 3 values aggregated from the table data.
    b. First one is “Count of users from Kiev or kiev” of numeric type.
    c. Second one is “Sum of ages of three oldest users from table” of numeric type.
    d. Third one is “Longest string of first name + last name pair” of string type.

Please, do not make things not described in items above.
If *data.json* file accidentally breaks, replace it with a fresh backup from *backup/data.json.*

Optional

If this task is too easy for you, create an “edit” button near each “delete” button and utilize PUT request in some way.

Layout Example

Table

First Name	Last Name	DOB	Location	Actions
Bryant	Castillon	01.01.2001	kiev	Edit Delete
Martin	Callender	01.01.2002	Kiev	Edit Delete

Summary

Count of users from Kiev or kiev
2
Sum of three oldest user ages
33
Longest string of first name + last name
Bryant Castillon

Form

First name
Last name
Date of birth mm/dd/yyyy
Location
Submit