Apartment marketplace application

Description

It's a simple web based application for managing renting apartments marketplace. The main idea is making a simple admin panel for the marketplace with a list of available apartments and minimal functionality around it. Users have the ability to sort and filter apartments, create new ads and remove existing.

The application consists of two parts: a backend part (Web API) and a frontend part (SPA).

Completion of one of these parts is enough to successfully pass the technical task. You can choose what you want to implement. Completing both parts would be a plus.

FrontEnd Part

TECHNOLOGIES:

- VanillaJS or one of these React /Angular/ Vue
- LocalStorage (use real API in case of both parts implemented)

TASK

Create a single page app for apartment marketplace. App user should be able to:

- View list of available apartments;
- Sort by price and filter by rooms;
- Add new apartment;
- Delete an apartment;

Apartment model:
- id: string
- rooms: number
- name: string
- price: number
- description: string

DESCRIPTION:

1. As a user, I should see the page which contains a list of available apartments and a form for adding new apartments. Available apartments list contains the list of apartments with the following fields:

    - Apartment name
    - Number of rooms
    - Price
    - Description

    Each element of the list should have a **delete** button. If I click this button, the apartment is removed from the list. Also, I should see the number of total available apartments.

2. Above the apartments list, I should see a **Sort by** a dropdown with the options following:
    - **Price - lowest to highest**
    - **Price - highest to lowest**

    When I select sorting mode, apartments are sorted by price accordingly.

3. Above the apartments list, I should see a **rooms** filter field (Text input or dropdown). When I enter/select the number of rooms, apartments are filtered by the number of rooms.

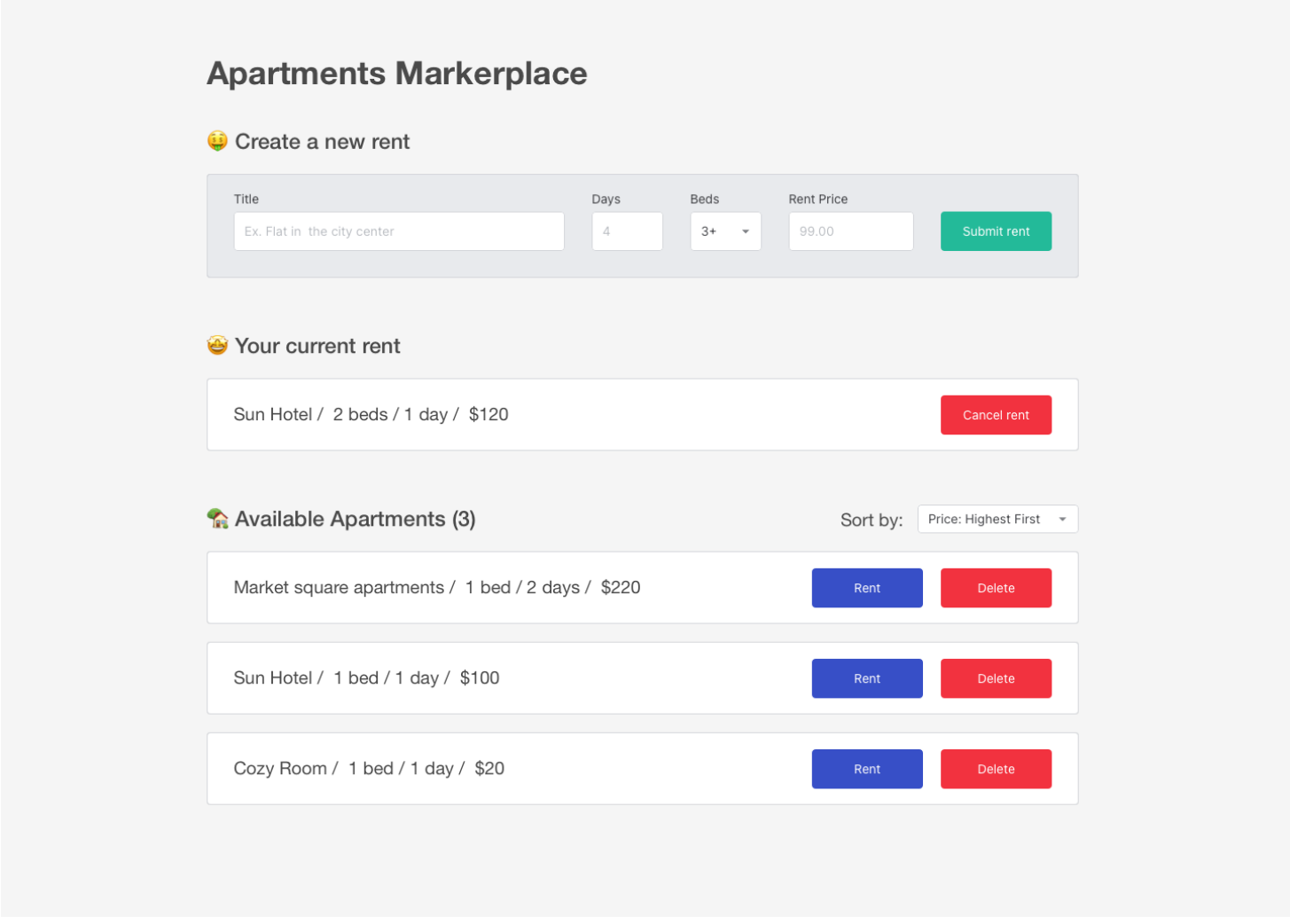
4. The page should contain a **form** for adding a new apartment. The **form** should consist of the following fields:
    - Apartment name
    - Number of rooms
    - Price
    - Description

When I click the **Add** button, an apartment with entered values is added to the list of apartments. The **form** has validation with the following rules:

- Apartment name - can't be empty. Max length is 99 characters
- Number of rooms - accepts only numbers. Can't be 0 or less than 0
- Price - accepts only numbers. Can't be 0 or less than 0
- Description - Optional text. Max length is 999 characters

5. Initial 4 apartments should be presented at application start. The apartment list should save its state, so I can close and reopen or reload the page, and still see changes I made. (Using the API if the Backend part of the task is also implemented, or using Localstorage in the browser)

UI example



Backend Part

TECHNOLOGIES:

- NestJs / Express
- PostgreSQL / MongoDB / Sqlite
- TypeORM / Mongoose / Sequelize (use ORM is optional)

TASK

Create a web API for apartment marketplace.
• ***GET /apartments*** [Get list of apartments; sorting (price) and filtering (rooms)];
• ***GET /apartments/{id}*** [Get apartment details];
• ***POST /apartments*** [Add new apartment; (validation rules: rooms-number, name-max length[99], price-number, description-max length[999])];
• ***DELETE /apartments/{id}*** [Delete apartment];

Would be a plus:
• ***PUT /apartments/{id}*** [Edit apartment details];

Apartment model:
- **id**: string
- **rooms**: number
- **name**: string
- **price**: number
- **description**: string

DESCRIPTION:

As a user, I should use the apartment marketplace API which implements basic CRUD operations.
I should get a list of apartments using the Get endpoint (GET /apartments). I can also use query parameters for sorting and filtering:

- price: string, two values: asc/desc
- rooms: number

For example:

http://localhost:8000/apartments?price=asc&rooms=2

I should get a specific apartment using the Get endpoint with specified Id (GET /apartments/{id}). For example.:

http://localhost:8000/apartments/3

I should delete a specific apartment I should delete using the DELETE endpoint with specified Id (DELETE /apartments/{id}.

I should add a new apartment using the POST endpoint (POST /apartments) and specifying body

parameters: rooms, price, name, and description. The body parameters should validate according to the following validation rules:

- rooms: number, value > 0
- price: number, value > 0
- name: string, value length < 99
- description: string, value length < 99

Body example:

```
{
  "rooms": 3,
  "name: "Amazing room near tower bridge",
  "price": 1650,
  "description": "This Room is located opposite Shadwell DLR station."
}
```

In case the body is invalid, I should see a response with a "400" status code.

Optional feature: I should update a specific apartment using the PUT endpoint (PUT/apartments/{id}). The body and its validation rules should be the same as at the "adding" endpoint.