Coding task Front-end dev

You are required to write a small building controls program in a backend application. This can be written in Java, Python or a backend JS framework such as NodeJS. The frontend application will then interact with the backend through an API. The Frontend should be a NextJS application utilizing React. You can include open source libraries, or testing frameworks if you choose to do so.

This application has the following requirements:

The Backend

- All **Rooms** should have a unique ID, the name of the occupant, a current temperature, a status for if heating is enabled (i.e. a heater), and a status for if cooling is enabled (i.e. an air conditioner).
  - Initially a room's temperature should be a random value between 10 °C and 40 °C.
- A **Building** can have 0 or more **Rooms**
  - A building should have a requested temperature for the whole building. The default requested temperature should be 20.0 °C
  - A building should have the ability to change the requested temperature
  - A building should have the ability to add additional Rooms
- Main Application:
  - A building should be created. Initially the building should have 4 Rooms. The building should also have a requested temperature of 25.0 °C.
  - The application should initially determine which rooms have heating or cooling enabled based on the following logic.
    - If the room temperature is below the requested building temperature, heating should be enabled.
    - If the room temperature is above the requested building temperature, cooling should be enabled.

The Frontend

- Be able to show detailed statuses of a Building and its rooms
  - Have a card or header that shows the properties of the Building
  - Be able to show each Room as a card with the properties, temperature values and heating/cooling status
- Have a method for the user to add, remove and edit rooms in a building
- Have the ability to set the requested temperature of a Building
  - Be able to show the updated cooling/heating statuses of the rooms based on the new temperature

Additional Notes

- The application should demonstrate server-side rendering, where the application logic will be handled in the backend and data queried through the server application and passed to the client
- There is no requirement to have persistent storage. All main application data can be held in memory and initialized to the default configuration on start.

Submitting

- Clone the provided repository (https://github.com/ci-keiran-hines/coding-challenge-room-ui) as a base for your implementation
- Show multiple commits for your progress
- In a readme state any build and run instructions, as well as any assumptions, including where you felt a particular requirement could be interpreted multiple ways.

Extras

- Allow the user to add/remove buildings
  - Have a dashboard that shows a summary of the status and properties for all Buildings
  - Have a method for the user to navigate between each building’s page
- Make the layout responsive
- Have all room temperature values update on a timer based on the rooms heating/cooling status. i.e. rooms that are being heated should increase in temperature gradually, and vice versa for rooms in cooling. The value and status changes should update the dashboards as well
- Demonstrate branching and merging with Git.