

REACT exercise :: Animal in the Forest

Description

Send your forest Animal on a Journey. And count the number of journeys.

With

```
// data structure of Animal
{
  id: String,
  type: String | null,
  journeysNo: Number,
  isHere: Bool,
}

// initial state of Animal
{
  id: nanoid(),
  type: null,
  journeysNo: 0,
  isHere: true,
}
```

Do

1. **FIRST**, create the UI with following elements:
 - `<div>` to display the Animal's type (`type`), number of journeys (`journeysNo`), and information about the Animal currently being (or not) in the Forest (`isHere`)
 - `<form>` with one input field (Animal's type) and one button (save Animal type)
 - `<button>` that says either `Go on a Journey` or `Return from the Journey` (depends on `isHere`)
2. **THEN**, write the application's logic:
 - when the user clicks the form's button, update the Animal's `type`
 - when the user clicks the journey button:
 - toggle `isHere`
 - (only on the start of a Journey) increment `journeysNo` by 1
3. **OPTIONAL**, ensure that the application has this structure:
 - `<App />` with the Animal's state
 - `<Animal />` with the UI
 - `<Controls />` with the `<form>` and `<button>` for updating the Animal's state
 - Pass data and functions down as props from `<App />` as needed.

Use

- `nanoid` npm package

- hook `useState`
- controlled or uncontrolled form (whichever you find suitable)
- conditional rendering