

Video: <https://d3dyfaf3iutrxo.cloudfront.net/general/upload/8470dc9dee0b401eb9f2135a952c51c0.mp4>

Project Requirements:

1. **Responsiveness:** The app should adapt to various screen sizes.
2. **Reusable Components:** Practice reusability in React code.
3. **Asynchronous Functionality and React Hooks:** Use async/await and React Hooks effectively.
4. **Styling:** Apply CSS to create an appealing user interface. We will also add a VERY FANCY heading.

This will be a step-by-step project. Follow these steps to complete the code:

Phase 1: Structure

1. App Structure:

- Create a main heading "Pokemon Kingdom".
- Construct the app-container, consisting of a pokemon-container to hold Pokemon cards, and a button to fetch and load more Pokemon cards.

2. API Interactions:

- Use the <https://content.newtonschool.co/v1/pr/64cce982071a9ad01d36ff6/pokemonspages1> to fetch initial Pokemon data.

This will return a list of 20 pokemons in the following format.

```
[
{
  "count": 10,
  "next":
  "https://content.newtonschool.co/v1/pr/64ccef632071a9ad01d36fea/pokemonspages2",
  "results": [
    {
      "name": "bulbasaur",
      "url": "https://content.newtonschool.co/v1/pr/64cc0d779d08d5aa491c7160/pokemons1"
    },
    {
      "name": "ivysaur",
      "url": "https://content.newtonschool.co/v1/pr/64cc111e9d08d5aa491c71e8/pokemons2"
    },
    {
      "name": "venusaur",
      "url": "https://content.newtonschool.co/v1/pr/64cc11569d08d5aa491c71fc/pokemons3"
    },
    {
      "name": "charmander",
      "url": "https://content.newtonschool.co/v1/pr/64cc119a9d08d5aa491c7209/pokemons4"
    },
    {
      "name": "charmeleon",
      "url": "https://content.newtonschool.co/v1/pr/64cc11d09d08d5aa491c7214/pokemons5"
    },
    {
      "name": "charizard",
      "url": "https://content.newtonschool.co/v1/pr/64cc11ff9d08d5aa491c7222/pokemons6"
    },
    {
```

```

    "name": "squirtle",
    "url": "https://content.newtonschool.co/v1/pr/64cc12479d08d5aa491c722e/pokemons7"
  },
  {
    "name": "wartortle",
    "url": "https://content.newtonschool.co/v1/pr/64cc127f9d08d5aa491c724d/pokemons8"
  },
  {
    "name": "blastoise",
    "url": "https://content.newtonschool.co/v1/pr/64cc12ce9d08d5aa491c725a/pokemons9"
  },
  {
    "name": "caterpie",
    "url": "https://content.newtonschool.co/v1/pr/64cc13009d08d5aa491c7267/pokemons10"
  }
]
}
]

```

IMPORTANT NOTE: The API returns "an array" of size 1. The actual API Response is that first element of the array.

- Make additional requests to fetch more detailed data from each Pokemon's specific URL.

- IMPORTANT: Each pokemon URL will also return an array of size 1. The actual JSON data would be the first element of the array.**[tip]:You can fetch and store these details in advance.**

3. Pokémon Cards:

- Each card, which can be a separate `PokemonThumbnail` component, should display the Pokemon's image, name, type, and a " `Know more...` " button.

4. Expanded Content Component: - Create an `ExpandedContent`` component that reveals more information about a Pokemon when the " `Know more...` " button on its card is clicked. This overlaying component should cover the cards, blurring the background and preventing clicks on the background.

- The `ExpandedContent`` component should also have a close button.

5. Load More Feature: - The "More Pokemons" button should fetch more Pokemon data from the API and append them to the existing cards.

- *Check the API Response carefully to observe how can you do this.*

Now our Structure is ready. Let's talk about styling of our project.

Phase 2 - Styling:

1. **Heading and Layout:** Look at the heading shown in the video and try to implement it using CSS. You can use grid or flexbox to keep it in the center.

2. **Card Styling:**

- Each Pokemon card should have a different background gradient depending on the Pokemon's type.

- The "Know more..." button on each card should have a contrast-enhanced gradient similar to its corresponding card.

3. **Responsive Design:** use CSS in such a way to ensure the website is responsive. Mainy cards and heading

4. **Hover Effects:** The cards should get a height elevation type effect on hover. The "Know more..." button's gradient should also change on hover.

5. **Expanded Content Styling:**

- Position the `ExpandedContent` component fixed in the center of the screen.
- Create a blurr effect on the background when the `ExpandedContent` card is open.

