

<https://github.com/Tomma252/ITM-Eksamen-Test.git>

Jeg har valgt å gå for vanskelighetsgrad D.

Tomma252 i GitHub = Thomas Berntsen

Jeg valgte å gjøre det slik at søker man på en pokémon kommer man direkte til siden som viser informasjon om søkt pokémon, ettersom navner er unike og trenger dermed ikke trykke flere ganger for å få informasjon om søkt pokémon. Dersom et søk er ugyldig blir bruker informert om dette.

En utfordring som jeg ikke klarte å løse var å få linjen med TYPE(S) til å være på samme nivå som navn på pokémon

BLASTOISE



TYPE(S)

water

STATS

hp: 79

attack: 83

defense: 100

special-attack: 85

special-defense: 105

speed: 78

En annen var at jeg startet prosjektet i den tro at jeg kunne arbeide mot karakterkrav C, så jeg gjorde noe klart for bruk av Sanity, men alas, klarte ikke dette og Sanity ble værende i prosjektet.

ChatGPT hjalp ved en feil på hvordan api kall ble skrevet i en <p>-tag:

It seems like the error is occurring because the ``ability.description`` object is undefined for some abilities. This issue likely arises when the API response does not include the expected structure for ability descriptions.

To fix this issue, you can add a check to ensure that ``ability.description`` exists before accessing its properties. Here's how you can modify the rendering part of your component to handle this:

```
javascript Kopier kode

{pokemon.abilities.map((ability, index) => (
  <section key={index}>
    <h4>{ability.name}</h4>
    <table>
      <tbody>
        <tr>
          <td>Effect:</td>
          <td>{ability.description ? ability.description.effect : 'No des
        </tr>
        <tr>
          <td>Short Effect:</td>
          <td>{ability.description ? ability.description.short_effect : '
        </tr>
      </tbody>
    </table>
  </section>
) ) )
```

By adding these conditional checks (``ability.description ? ... : 'No description available.'``), you ensure that the component won't attempt to access properties of ``ability.description`` if it's undefined, thus preventing the `TypeError`. Instead, it will display a fallback message when the description is not available.

Make sure to adjust the logic and styling as needed to fit your UI requirements.

For å få dette svaret kopierte og limte jeg inn feilmeldingen fra console:

```
Uncaught TypeError: Cannot read properties of undefined (reading 'effect')
```

Jeg spurte ChatGPT om et eksempel på hvordan man kan bruke pokeapi linken til å generere noen enkle api-kall og fikk dette:

```
jsx Kopier kode

import React, { useState } from 'react';
import axios from 'axios';

const SearchResult = () => {
  const [search, setSearch] = useState('');
  const [results, setResults] = useState([]);

  const handleSearch = async () => {
    try {
      const response = await axios.get(`https://pokeapi.co/api/v2/pokemon/${search}`);
      setResults([response.data]);
    } catch (error) {
      console.error("Pokémon not found!");
      setResults([]);
    }
  };

  return (
    <div>
      <h2>Search for a Pokémon</h2>
      <input type="text" value={search} onChange={(e) => setSearch(e.target.value)} />
      <button onClick={handleSearch}>Search</button>
      <div>
        {results.length > 0 && results.map((pokemon, index) => (
          <div key={index}>
            <h3>{pokemon.name}</h3>
            <p>Height: {pokemon.height}</p>
            <p>Weight: {pokemon.weight}</p>
            <p>Base experience: {pokemon.base_experience}</p>
          </div>
        ))}
      </div>
    </div>
  );
};

export default SearchResult;
```

Dette dannet grunnlaget for prosjektet, sammen med arbeidskrav fra tidligere i semesteret.

Når jeg ikke klarte å få til at type lå på samme linje som navn på pokémon, henviste jeg meg til ChatGPT for hjelp og endte opp med å bruke deler av dette:

```
css
Kopier kode

.pokemon {
  width: 100%;
  padding: 40px 20px;
  margin-bottom: 20px;
  display: flex; /* Changed from grid to flex */
  flex-wrap: wrap; /* Allows items to wrap to the next line if needed */
  gap: 20px;
}

.pokemon__name {
  font-size: 24px;
  margin-bottom: 10px;
  width: 100%; /* Ensures it takes up the full width */
}

.pokemon__image {
  width: 400px;
  height: 400px;
}

.pokemon__details {
  flex: 1; /* Allows it to grow and take up available space */
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  margin-left: 20px; /* Added spacing between name and details */
}

.pokemon__types, .pokemon__stats {
  font-size: 16px;
  margin-bottom: 10px;
  list-style: none;
}

.pokemon__types span, .pokemon__stats li {
  margin-bottom: 5px;
}

.pokemon__abilities {
  width: 100%; /* Ensures it takes up the full width */
  margin-top: 20px;
}

.pokemon__abilities-name {
  font-size: 20px;
  margin-bottom: 5px;
}

.pokemon__abilities-description {
  font-size: 14px;
}
```

Melding ChatGPT

