Analiza komponentu Excuse z metaforami

1. Wprowadzenie
Komponent `Excuse` to interaktywny element Reacta, który dzia■a jak magiczna skrzynka na wymówki. Jego zadaniem jest:
- odbieranie nowych danych od u■ytkownika,
- tworzenie i zapami■tywanie wymówek,
- przechowywanie ich w pami■ci przegl■darki (localStorage),
- oraz umo■liwienie edycji i usuwania.

2. JSON.stringify i JSON.parse
Wyobra■ sobie, ■e chcesz przechowa■ ca■■ ksi■■k■ (czyli dane) w pude■ku (czyli localStorage). Nie mo■esz tam w■o■y■ prawdziwej ksi■■ki — musisz j■ zamieni■ w tekst.
- JSON.stringify(dane) – zamienia dane (np. obiekt) na tekst do przechowania.
- JSON.parse(tekst) – zamienia tekst z powrotem na dane.

W kodzie:
localStorage.setItem('userExcuses', JSON.stringify(userExcusesList));
const savedExcuses = localStorage.getItem('userExcuses');
return savedExcuses ? JSON.parse(savedExcuses) : [];

3. Kluczowe zmienne i ich rola
- acceptData – nowe dane wej■ciowe.
- userExcusesList – lista zapisanych wymówek.
- lastExcuse – ostatni dodany wpis.
- excuseLevel – poziom wyliczany z ró■nicy credibility - creativity.
- excuseText – losowa wymówka z odpowiedniej kategorii.
- excuses – globalna baza wymówek.

4. Dzia■anie komponentu krok po kroku
1. Wczytanie danych z localStorage (useState).
2. Generowanie nowej wymówki przy zmianie acceptData (useEffect).
3. Zapis do localStorage.
4. Edycja i usuwanie wpisów.
5. Wy■wietlanie na ekranie (z podzia■em na pilne i zwyk■e).

5. Metaforyczne porównania
- useState – pami■■ skrzynki.
- useEffect – laboratorium.
- localStorage – sejf.
- excuseLevel – balans kreatywno■■ vs realizm.
- edycja/usuwanie – praca z kartkami na tablicy og■osze■.

6. Na co zwróci■ uwag■ przy pytaniach nauczyciela
- Ró■nica JSON.stringify vs JSON.parse.
- Dzia■anie useEffect.
- Sk■d i dok■d id■ dane.
- localStorage jako trwa■a pami■■.
- Edycja/usuwanie – jak dzia■a logika.