# Kvíz Alkalmazás (React)

## Cél:

A cél egy React alkalmazás fejlesztése, amely lehetővé teszi a felhasználók számára, hogy egy kvízt töltsenek ki, beküldjék válaszaikat és visszajelzést kapjanak az eredményeikről. Az alkalmazás az alábbi API végpontokat használja.

## API Végpontok

### GET /quiz

Lekérdezi a kvíz kérdéseket és a hozzájuk tartozó válaszokat.

Példa válasz:

[  
 {  
 "id": 1,  
 "text": "Mi Japán fővárosa?",  
 "answers": [  
 { "id": 101, "text": "Tokió" },  
 { "id": 102, "text": "Osaka" },  
 { "id": 103, "text": "Kiotó" },  
 { "id": 104, "text": "Nagoya" }  
 ]  
 }  
 ]

### POST /validate

Ellenőrzi a felhasználó válaszait és visszaküldi az eredményt.

Egyetlen válasz validációjára is alkalmazható, többet egyszerre.

Példa kérés:

[  
 { "questionId": 1, "answerId": 101 },  
 { "questionId": 2, "answerId": 202 }  
 ]

Példa válasz:

{  
 "correct": 1,  
 "total": 2,

“details”:{

{

questionId:1,

correct:true

},

{

questionId:1,

correct:false

}

}

}

## A kezdéshez:

A feladat “backend” alkönyvtárában ”npm install”, aztán “npm start” utasítás kiadása után a backend elindul. A webes swagger dokumentáció a <http://localhost:3000/api-docs> linken elérhető.

A react vite alap projekt már tartalmazza a megfelelő proxybeállításokat, a kéréseket a “/api” felé kell indítani.

Az adatbázis alapértelmezetten tartalmaz adatokat, nem szükséges feltölteni tesztadatokkal.

<https://react.dev/>

<https://mui.com/>

<https://axios-http.com/docs/intro>

A react projekt szintén npm install után, az “npm run dev”-el indítható.

## Feladatok:

1. **Szükséges modulok telepítése (1p)**

NPM install segítségével telepítsd a feladat megoldásához szükséges modulokat. Axios, npm stb.

1. **Kérdések lekérdezése, tárolása rendereléshez megfelelő módon. (2p)**

Indíts HTTP GET kérést a /api/quiz URL-re. És tárold az adatokat a megfelelő react hookokat használva

1. **Kérdések lépésről lépésre megválaszolása. (3p)**

Vezesd végig a felhasználót a kérdések sorozatán, egyszerre egy kérdést és annak válaszait megjelenítve. A kérdés megválaszolása után a jelenítsd meg a következő kérdést.

1. **Válaszok ellenőrzése menet közben (1p)**

A /api/validate URL-t használva már az egyes kérdések megválaszolása után is vissza jelezhetsz a felhasználónak a válasza helyességét illetően.

1. **Válaszok ellenőrzése a quiz végén. (2p)**

A teljes quiz kitöltése után értékeld ki a felhasználó válaszait. Jelenítsd meg a ponteredményt.

1. **Reszponív kinézet (2p)**

A kész megoldás mobil készüléken is megfelelően jelenítse meg a tartalmat

1. **Kód/Projektstruktúra (1p)**

Legyen a projekt forráskódja értelmezhető, jól szervezett. Hasonló képp a projekt felépítése is tartalmazza a megfelelő szintű komponensekre bontást.

## Leadás:

Legkésőbb 17:30-ig adjátok le a megoldást, ellenben a feladat nem értékelhető.

A következő lépéseket kérném a feltöltésnél:

A projekt könyvtárát úgy ahogy megkaptátok, ugyan olyan tömörített formátumban vissza feltölteni a helyes megoldás tartalmával. Ehhez persze a gyökérkönyvtárban a frontend node\_modules könyvtárát és eggyel bentebb, a backend node\_modules könyvtárát is törölnötök kell!

Kérnék mindenkit hogy 2x-3x is ellenőrizze a megoldásának helyességét, a feltöltött fájl épségét, a feltöltés sikerességét.