

Komplex feladat megoldása

Backend

Feladat: Hozzon létre egy „naptar” nevű adatbázist, amely az alábbi táblákkal rendelkezik:

Felhasználók:

Mező	Tulajdonság
id	számláló
nev	szöveg, hossz: 50
email	szöveg, hossz: 100
created_at, updated_at	időbélyegek

Események:

Mező	Tulajdonság
id	számláló
szervezo	idegen kulcs
cím	szöveg, hossz: 255
leiras	szöveg, hossz: 65535 (text())
kezdes	dátum
befejezes	dátum
helyszin	szöveg, hossz: 255
tipus_id	idegen kulcs
created_at, updated_at	időbélyegek

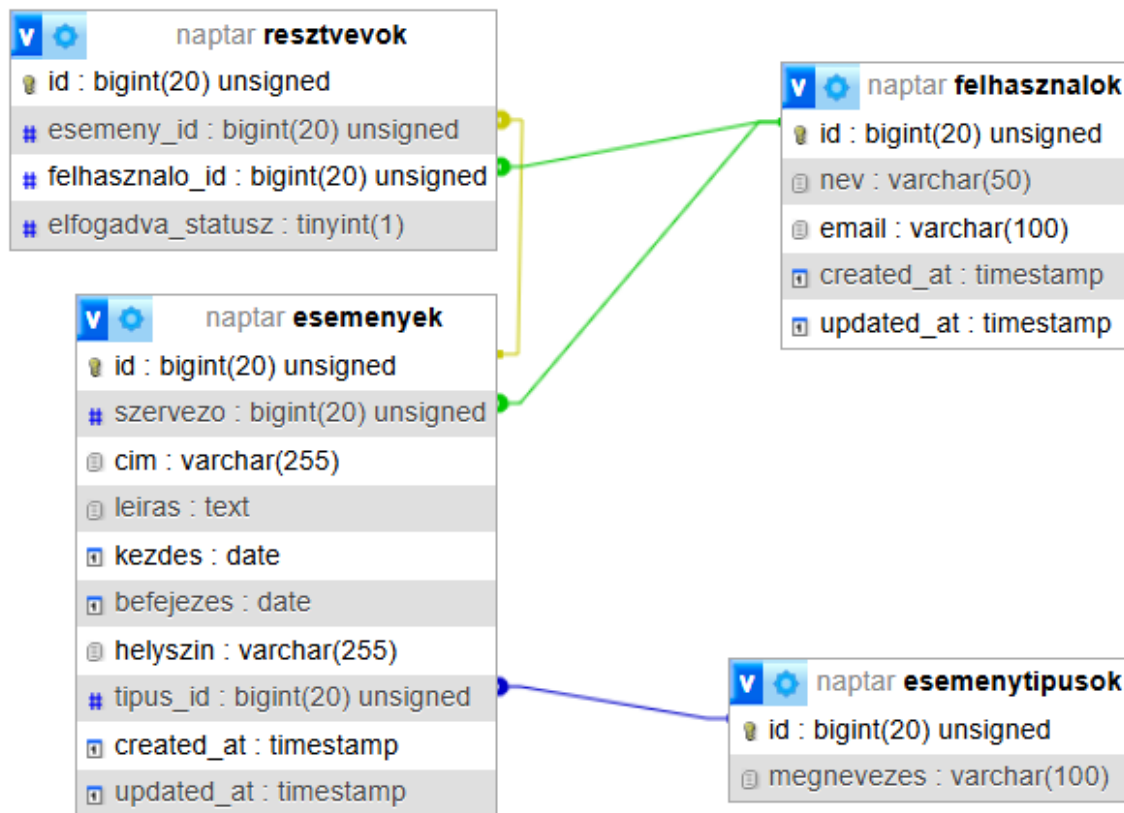
EsemenyTipusok:

Mező	Tulajdonság
id	számláló
megnevezes	szöveg, hossz: 100

Resztvevok:

Mező	Tulajdonság
id	számláló
esemeny_id	idegen kulcs
felhasznalo_id	idegen kulcs
elfogadva_statusz	boolean

Kapcsolatokat bemutató ábra:



Feladat: Töltse fel adatokkal a táblákat az alábbi táblázatoknak megfelelően!

Felhasznalok:

nev	email
Példa Péter	pelda.peter@email.hu
Metódus Mária	metodus.mario@email.hu

Események:

szervezo	cim	leiras	kezdes	befejezes	helyszin	tipus_id
Példa Péter (id)	Tenisz	1 órás teniszedzés	2025-04-01	2025-04-01	Sportpálya	Sportolás (id)
Metódus mária (id)	Továbbképző értekezlet	2 napos értekezlet	2025-04-03	2025-04-05	Budapest, XY konferenciaterem	Meeting (id)

EsemenyTipusok:

megnevezes
Születésnap
Meeting
Koncert

Sportolás

Resztvevok:

esemeny_id	felhasznalo_id	elfogadva_statusz
1	1	0
2	2	1

Feladat: Készítsen a „Felhasznalok” nevű táblához végpontot, amely az összes adatot megjeleníti! A végpont a „/api/felhasznalok” aloldalon legyen elérhető!

Feladat: Készítsen az „Esemenyek” táblához végpontot, amely adatok feltöltésére szolgál! Feltételezzük, hogy az adatok formátuma helyes, így ellenőrzéssel nem kell foglalkoznia! A végpont a „/api/esemenyFeltoltes” aloldalon legyen elérhető!

- Ahhoz, hogy az adatfeltöltés sikerességét tudja ellenőrizni is, a http kéréshez tartozó függvény legyen get()!
- A feltöltött adatok tetszőlegesek!

Feladat: Készítsen az „Esemenyek” táblához végpontot, amely a URL-ben megadott *id* alapján frissít egy rekordot a táblából! A végpont a „/api/esemenyFrissites” aloldalon legyen elérhető!

- Adatfrissítés után térjen vissza az oldal 200-as státuszkóddal!
- a frissített rekord tetszőleges, és tetszőleges adattal frissíthető!
- Ahhoz, hogy az törlés sikerességét tudja ellenőrizni is, a http kéréshez tartozó függvény legyen get()!

Frontend

Készítsen egy React alkalmazást, ami a korábban létrehozott Backend projekthez kapcsolódik, és az alábbi komponensekkel rendelkezik:

- Header: A komponensben készítsen egy <H1> elemet, amely megjeleníti a komponensnek átadott szöveget! A komponens az App.jsx fájlban kerüljön felhasználásra, itt adja át a komponensnek az „Naptáralkalmazás” szöveget!
- Main: A komponens betölt egy tetszőlegesen választott háttérképet, továbbá leírja a webalkalmazás célját/leírását/működését röviden!
- EsemenyFeltoltes: A komponens egy űrlapot valósít meg, amely új esemény létrehozását teszi lehetővé. Ehhez kérjen be minden szükséges adatot, a szervező és típus esetén jelenleg elég, ha számozatosítót ad meg!

- **EsemenyFrissites:** A komponens egy űrlapot valósít meg, amely meglévő esemény módosítását teszi lehetővé. Az frissítendő eseményt azonosító alapján határozzuk meg!