**Eventhost**

2023-2024

Pásztor István: Frontend

Szeri Bendegúz: Test

Takács Dániel: Backend

Tartalom

[Projekt célja: 4](#_Toc164075099)

[Felhasznált technológiák: 5](#_Toc164075100)

[Követelmények: 6](#_Toc164075101)

[Specifikáció: 7](#_Toc164075102)

[Frontend: 8](#_Toc164075103)

[AboutUs 8](#_Toc164075104)

[Company 8](#_Toc164075105)

[Company/Update 9](#_Toc164075106)

[Events 10](#_Toc164075107)

[Events/[id==uuid] 11](#_Toc164075108)

[Events/[id=uuid]/update 12](#_Toc164075109)

[Home 13](#_Toc164075110)

[Organizer/[id=uuid]/events/create 14](#_Toc164075111)

[Profile/Organizer/[id=uuid] 15](#_Toc164075112)

[Profile/User/[id=uuid] 17](#_Toc164075113)

[Profile/User/[id=uuid]/update 18](#_Toc164075114)

[Registration 19](#_Toc164075115)

[MainPage 20](#_Toc164075116)

[Backend: 23](#_Toc164075117)

[AboutUs 23](#_Toc164075118)

[Company 24](#_Toc164075119)

[Company/[id=uuid]/update 25](#_Toc164075120)

[Events 26](#_Toc164075121)

[Events/[id=uuid] 27](#_Toc164075122)

[Events/[id=uuid]/update 28](#_Toc164075123)

[Home 30](#_Toc164075124)

[Organizers/[id=uuid]/events/create 31](#_Toc164075125)

[Profile/organizer/[id=uuid] 32](#_Toc164075126)

[Profile/organizer/[id=uuid]/update 33](#_Toc164075127)

[Profile/user/[id=uuid] 34](#_Toc164075128)

[Profile/user/[id=uuid]/update 35](#_Toc164075129)

[Registration 37](#_Toc164075130)

[MainPage 38](#_Toc164075131)

[Adatbázis: 40](#_Toc164075132)

[Tervek: 42](#_Toc164075133)

# Projekt célja:

A projektünk célja az hogy a lehető legkönnyebben lehessen saját privát rendezvényeket létrehozni és azokra a résztvevőket meghívni. Ez a felhasználóknak azért lesz igazán kedvező mert a könnyen kezelhető felület segítségével a program betanulása nélkül is másodpercek alatt létre tudnak hozni egy eseményt.

# Felhasznált technológiák:

Tailwind

Skeleton.dev

Svelte

EdgeDb

Edgeql-js

TypeScript

# Követelmények:

Felhasználó létrehozás, esemény létrehozás, bejelentkezés, külön oldal az eseményeknek az események létrehozásának a bejelentkezésnek.

# Specifikáció:

*Regisztráció és Bejelentkezés:*

A felhasználóknak regisztrálniuk kell egy fiókot a weboldalon.

A regisztráció során meg kell adniuk a személyes információikat, például nevet, e-mail címet stb.

Miután regisztráltak, be kell jelentkezniük a fiókjukba.

*Főoldal:*

A főoldalon a felhasználók áttekintést kaphatnak az aktuális és közelgő eseményekről.

Lehetőség van új események létrehozására a "Új esemény létrehozása" gombra kattintva.

*Események Oldal:*

Az "Események" oldalon láthatók az összes rendelkezésre álló események.

Minden esemény nevére kattintva részletes információkat láthatnak az eseményről.

Az esemény oldalán lehetőség van részvételi szándék kifejezésére a "Részvétel" gomb segítségével.

*Esemény Létrehozása:*

A "Új esemény létrehozása" gombra kattintva a felhasználók kitöltik az esemény részleteit (cím, időpont, helyszín stb.).

A létrehozott eseményen belül lehetőség van meghívottak hozzáadására, akiket a rendszer QR kóddal azonosít.

Az esemény létrehozásakor a rendszer generál egy egyedi QR kódot az eseményhez.

*Meghívások Elfogadása:*

A meghívottak bejelentkezhetnek a programba.

A "Meghívások" részen láthatják azokat az eseményeket, amelyekre meghívták őket.

Elfogadhatják vagy elutasíthatják a meghívásokat.

*QR Kód Beolvasása:*

Az esemény napján a résztvevők be tudják olvasni a meghívó QR kódját az esemény részleteinek megtekintéséhez és részvételéhez.

# Frontend:

## AboutUs

**Navigációs sáv**

A weboldal tetején található a navigációs sáv, amely tartalmazza az Eventhost logót, a belépési linkeket, valamint az oldalak közötti navigációs hivatkozásokat. A navigációs sáv reszponzív, ami azt jelenti, hogy az eszköz méretétől függően változtatja a megjelenését.

**Bemutatkozás szekció**

A navigációs sáv alatt található a "Bemutatkozás" szekció, amely bemutatja az Eventhost csapat tagjait. Ez a szekció három Svelte komponensből áll, amelyek mindegyike egy csapattagot jelenít meg. Minden csapattag komponens tartalmaz egy képet, a tag nevét, egy rövid bemutatkozó szöveget.

**Belépés**

A weboldal a SvelteKit signIn függvényét használja a felhasználók Google fiókkal történő bejelentkezéséhez. Ha a felhasználó be van jelentkezve, a navigációs sávban megjelenik a felhasználó e-mail címe és a kijelentkezés lehetősége.

## Company

**Importálások:**

PageData típus: Ez meghatározza a komponens által elvárt adatok struktúráját.

page a $app/stores-ból: Ez hozzáférést biztosít az aktuális oldal adataihoz.

signIn és signOut az @auth/sveltekit/client-ből: Ezeket a függvényeket a Google-fiókkal történő bejelentkezéshez és kijelentkezéshez használják.

**Adatok:**

data: Ez a változó tárolja az oldal adatait.

**Sablon:**

A komponens egyetlen HTML-elemet használ, amelynek data-theme attribútuma "wintry"-re és az osztálya "w-screen h-screen overflow-y-auto" értékre van állítva.

**Fejléc:**

A fejléc szekció tartalmaz egy navigációs sávot az Eventhost logóval és linkekkel.

Megjeleníti a felhasználó adatait és a bejelentkezési/kijelentkezési funkciókat is a bejelentkezési állapot alapján.

**Fő tartalom:**

A fő tartalmi terület a data változóban lekérdezett céges adatokat jeleníti meg. Ez magában foglal olyan részleteket, mint a szervezői kód, a leírás, a cím, a webhely és a közösségi média linkek.

Feltétel szerint, ha a felhasználó szervező, akkor megjelenik egy "Cég frissítése" gomb, amely egy cégfrissítő oldalra vezet.

## Company/Update

**Importálások:**

PageData típus: Ez meghatározza a komponens által elvárt adatok struktúráját.

page az $app/stores-ból: Ez biztosítja a hozzáférést az aktuális oldal adataihoz.

signIn és signOut az @auth/sveltekit/client-ből: Ezeket a függvényeket a Google-fiókkal történő bejelentkezéshez és kijelentkezéshez használják.

**Adatok:**

data: Ez a változó tárolja a lekérdezett céges adatokat.

**Sablon:**

**Fejléc:**

Eventhost logó és hivatkozások a navigációhoz.

Felhasználói adatok és bejelentkezési/kijelentkezési funkciók a bejelentkezési állapot alapján.

**Fő Tartalom:**

**Űrlap:** Ez az űrlap lehetővé teszi a szervezőknek, hogy frissítsék a céges adataikat. Az alábbi mezőket tartalmazza:

Név

EU adószám

Ország

Város

Irányítószám

Cím

Címrészletek

Weboldal

Közösségi média linkek (Facebook, Instagram, Linkedin, Tiktok)

Leírás

Rejtett mezők a céges e-mail címhez és a szervezői kódhoz ( a felhasználói adatok alapján előre kitöltve)

**Mentés gomb:** Az űrlap elküldése a céges adatok frissítésére.

## Events

**Importálások:**

Ezek a sorok a szükséges komponenseket és adatstruktúrákat importálják.

PageData típus: Ez meghatározza a komponens által elvárt adatok struktúráját.

page az $app/stores-ból: Ez biztosítja a hozzáférést az aktuális oldal adataihoz.

signIn és signOut az @auth/sveltekit/client-ből: Ezeket a függvényeket a Google-fiókkal történő bejelentkezéshez és kijelentkezéshez használják.

**Adatok:**

export let data: PageData; - Ez a sor egy data nevű változót deklarál PageData típusként. Ez a változó tárolja majd a lekérdezett eseményekről és esetlegesen a szervezőről szóló adatokat.

**HTML Szerkezet:**

Ez a rész definiálja az oldal HTML struktúráját.

Fejléc: Ez tartalmazza az Eventhost logót, a navigációs linkeket és a felhasználói információkat a bejelentkezési állapot alapján.

Törzs:

Tartalom: Ez a rész az események listáját jeleníti meg.

Végigiterál a data.event tömbön (feltételezve, hogy létezik) és megjeleníti az egyes eseményekre vonatkozó információkat, többek között a címet, a helyszínt, a szervezőt, a leírást és egy linket az esemény részletei oldalára.

Ha nem talál eseményt, a "Jelenleg nincs eseményed" üzenet jelenik meg.

Gomb: Ez a gomb lehetővé teszi a szervezők számára új esemény létrehozását (csak akkor látható, ha létezik szervezői adat).

**Feltételes Logika:**

Ezek a szekciók feltételes utasításokat ({#if}) használnak a rendelkezésre álló adatok alapján történő tartalom megjelenítéséhez.

A felhasználói információkat és a Bejelentkezés/Kijelentkezés gombokat a bejelentkezési állapot ($page.data.session?.user) alapján jeleníti meg.

A "Létrehozás esemény" gombot csak akkor jeleníti meg, ha van szervezői adat (data.organizer).

## Events/[id==uuid]

**Szkript:**

Importál szükséges komponenseket (PageData) és adat-tárolókat (page). Ezen kívül importál felhasználói bejelentkezési és kijelentkezési függvényeket Google-autentikációval (signIn, signOut).

Deklarál egy data nevű változót PageData típusként. Ez a változó tárolja majd a lekérdezett eseményadatokat.

Kinyeri az esemény kezdő és befejező dátumát a data.event objektumból (feltételezve, hogy létezik), és a customDateFormatA és customDateFormatB függvények segítségével formázott sztringgé alakítja őket.

Hasonlóan a fentiekhez, további eseményadatokat nyer ki, mint például a helyszín, a leírás és a kezdő időpont.

**HTML:**

Fejléc szekció (header), amely tartalmazza az Eventhost logót, a navigációs linkeket és a felhasználói információkat a bejelentkezési állapot alapján.

Fő tartalom szekció (section), amely az esemény részleteit jeleníti meg:

Esemény címe (<h1>)

Helyszín részletek, beleértve a helynevet, országot, várost, irányítószámot, címet és egyéb cím részleteket (<p>)

Esemény dátumok formázott sztringben (<p>)

Esemény kezdő időpontja (<p>)

Esemény leírása (<p>)

Esemény frissítése gomb (ha létezik szervezői adat) (<a>)

## Events/[id=uuid]/update

**Komponensek**

**Script (TypeScript):**

Importál szükséges elemeket: PageData komponens az eseményadatokhoz, page store a globális állapot kezeléséhez, signIn és signOut függvények a Google-autentikációhoz.

Létrehoz egy data változót a PageData típusból az eseményadatok tárolására.

Kinyeri az esemény adatait a data objektumból, beleértve a címet, a helyszín részleteit (név, ország, város, irányítószám, cím, címrészletek), a kezdő és befejező dátumokat (egyéni dátumformázó függvényekkel formázva), a kezdő időpontot és a leírást.

Definiálja a customDateFormatA és customDateFormatB függvényeket az YYYY-MM-DD formátumú dátumozáshoz.

**HTML:**

**Fejléc (**header**):**

Eventhost logó

Navigációs linkek

Bejelentkezett felhasználó adatai (email cím) vagy bejelentkezési lehetőségek

**Fő Tartalom (**section**):**

Eseményinformáció-frissítő űrlap:

Esemény címe beviteli mező

Helyszín részletek beviteli mezői (név, ország, város, irányítószám, cím, címrészletek)

Dátumválasztók a kezdő és befejező dátumokhoz

Kezdő időpont beviteli mező

Esemény leírása szövegdoboz

"Mentés" gomb az eseményadatok frissítéséhez

## Home

**Szkript:**

Importál szükséges elemeket és függvényeket:

A PageData interface meghatározza az alkalmazásnak átadott adatok struktúráját.

A $app/stores-ból származó page  információkat tartalmaz az aktuális oldalról.

signIn és signOut függvények a felhasználói hitelesítéshez ( az Auth0 Sveltekit használatával).

onMount életciklus-hook a komponens betöltése utáni műveletekhez.

data változó az oldaltartalom tárolására ( a page store-ból töltődik be).

scrollToDiv függvény a "yourDivId" azonosítójú elemhez való gördítésre.

Az onMount hook a komponens betöltésekor meghívja a scrollToDiv függvényt, hogy az elemhez görgessen.

**Test:**

Beállítja a témát "téli"-re és engedélyezi a gördülékeny görgetést.

Fő konténer snap görgetési osztályokkal.

**Fejléc:**

Navigációs sáv az Eventhost logóval és linkekkel.

Feltételes tartalom a felhasználói bejelentkezési állapot alapján:

Bejelentkezett felhasználó: profilkép, e-mail cím és kijelentkezés gomb.

Be nem jelentkezett felhasználó: bejelentkezés gomb.

Hamburger menü gomb a mobilos navigációhoz.

**Tartalom:**

Háttérkép elmosódító szűrővel.

Központi részlet egy nagy címmel és leírással az Eventhost számára.

## Organizer/[id=uuid]/events/create

**Szkript:**

Importál szükséges elemeket és függvényeket:

A PageData interface meghatározza az alkalmazásnak átadott adatok struktúráját.

A $app/stores-ból származó page  információkat tartalmaz az aktuális oldalról.

signIn és signOut függvények a felhasználói hitelesítéshez ( az Auth0 Sveltekit használatával).

Az enhance függvény a $app/forms-ból segít az űrlap validálásában és beküldésében.

data változó az oldaltartalom tárolására ( a page store-ból töltődik be).

Logika a felhasználó e-mail címéből a domain név kinyerítésére és a splitEmail változóban való tárolására.

**Test:**

Beállítja a témát "téli"-re és engedélyezi a gördülékeny görgetést.

Fő konténer sötét háttérrel.

**Fejléc:**

Navigációs sáv az Eventhost logóval és linkekkel.

Feltételes tartalom a felhasználói bejelentkezési állapot alapján:

Bejelentkezett felhasználó: profilkép (ha elérhető), e-mail cím és kijelentkezés gomb.

Be nem jelentkezett felhasználó: bejelentkezés gomb.

Hamburger menü gomb a mobilos navigációhoz.

**Tartalom:**

Űrlap esemény létrehozásához:

Mezők az esemény részleteinek megadásához, mint például a cím, a helyszín (helynév, ország, város, irányítószám, cím, címadatok), a kezdő és befejező dátum, a kezdő óra és a leírás.

Egy rejtett mező rögzíti a felhasználó e-mail címéből származó domain nevet (korábban kinyerítve).

Egy "Mentés" gomb az űrlap elküldéséhez.

## Profile/Organizer/[id=uuid]

**Szkript:**

Importál szükséges elemeket és függvényeket:

A PageData interface meghatározza az alkalmazásnak átadott adatok struktúráját.

A $app/stores-ból származó page  információkat tartalmaz az aktuális oldalról, beleértve a munkamenet adatait is.

signIn és signOut függvények a felhasználói hitelesítéshez ( az Auth0 Sveltekit használatával).

data változó az oldaltartalom tárolására ( a page store-ból töltődik be).

**Test:**

Beállítja a témát "téli"-re és egy szűrőt használ a háttér elmosódításához.

Fő rész sötét háttérrel:

Megjeleníti a szervező nevét, ha az elérhető a data objektumban.

Megmutatja a szervező elérhetőségi adatait: telefonszám, ország, város, irányítószám, cím és címadatok a data objektumból.

Ha a bejelentkezett felhasználó a szervező, egy gomb segítségével frissítheti a profilját.

**Fejléc:**

az Eventhost logót és a navigációs sávot jeleníti meg.

Bejelentkezett felhasználók számára profilképet és e-mail címet jelenít meg kijelentkezési gombbal.

Be nem jelentkezett felhasználóknak bejelentkezési gombot kínál.

Lehet, hogy tartalmaz egy hamburger menü gombot a mobilos navigációhoz.

Profile/Organizer/[id=uuid]/update

**Szkript:**

Importál szükséges elemeket és függvényeket:

A PageData interface meghatározza az alkalmazásnak átadott adatok struktúráját.

A $app/stores-ból származó page  információkat tartalmaz az aktuális oldalról, beleértve a munkamenet adatait is.

signIn és signOut függvények a felhasználói hitelesítéshez ( az Auth0 Sveltekit használatával).

data változó az oldaltartalom tárolására ( a page store-ból töltődik be).

A szervező adatait (név, elérhetőségi adatok stb.) a data.organizer objektumból különálló változókba bontja a könnyebb kezelhetőség érdekében.

Az e-mail információt a data.organizer objektumból kinyeri (feltételezve, hogy az e-mail a szervezői adatok részét képezi).

**Fejléc:**

az Eventhost logót és a navigációs sávot jeleníti meg.

Bejelentkezett felhasználók számára profilképet és e-mail címet jelenít meg kijelentkezési gombbal.

Be nem jelentkezett felhasználóknak bejelentkezési gombot kínál.

Lehet, hogy tartalmaz egy hamburger menü gombot a mobilos navigációhoz.

**Test:**

Beállítja a témát "téli"-re és egy szűrőt használ a háttér elmosódításához.

Fő rész sötét háttérrel:

Megjeleníti a szervező nevét, ha az elérhető a data objektumban.

Megmutatja a szervező elérhetőségi adatait: telefonszám, ország, város, irányítószám, cím és címadatok a data objektumból.

Űrlapot jelenít meg beviteli mezőkkel az összes elérhetőségi adat szerkesztéséhez.

Rejtett beviteli mezők tárolják a data-ból lekérdezett szervezői kódot és e-mail címet. Ezeket előre kitöltik, és a felhasználó nem szerkesztheti őket.

"Mentés" gomb a frissített információk beküldéséhez ( POST kérés használatával).

## Profile/User/[id=uuid]

**Szkript:**

Importál szükséges elemeket és függvényeket:

A PageData interface meghatározza az alkalmazásnak átadott adatok struktúráját.

A $app/stores-ból származó page  információkat tartalmaz az aktuális oldalról, beleértve a munkamenet adatait is.

signIn és signOut függvények a felhasználói hitelesítéshez ( az Auth0 Sveltekit használatával).

data változó az oldaltartalom tárolására ( a page store-ból töltődik be).

**Test:**

Beállítja a témát "téli"-re és egy szűrőt használ a háttér elmosódításához.

Fő rész sötét háttérrel:

Megjeleníti a szervező nevét, ha az elérhető a data objektumban.

Megmutatja a szervező elérhetőségi adatait: telefonszám, ország, város, irányítószám, cím és címadatok a data objektumból.

Ha a bejelentkezett felhasználó a szervező, egy gomb segítségével frissítheti a profilját.

**Fejléc:**

az Eventhost logót és a navigációs sávot jeleníti meg.

Bejelentkezett felhasználók számára profilképet és e-mail címet jelenít meg kijelentkezési gombbal.

Be nem jelentkezett felhasználóknak bejelentkezési gombot kínál.

Lehet, hogy tartalmaz egy hamburger menü gombot a mobilos navigációhoz.

## Profile/User/[id=uuid]/update

**Szkript:**

Importál szükséges elemeket és függvényeket:

A PageData interface meghatározza az alkalmazásnak átadott adatok struktúráját.

A $app/stores-ból származó page  információkat tartalmaz az aktuális oldalról, beleértve a munkamenet adatait is.

signIn és signOut függvények a felhasználói hitelesítéshez ( az Auth0 Sveltekit használatával).

data változó az oldaltartalom tárolására ( a page store-ból töltődik be).

A szervező adatait (név, elérhetőségi adatok stb.) a data.user objektumból különálló változókba bontja a könnyebb kezelhetőség érdekében.

Az e-mail információt a data.user objektumból kinyeri (feltételezve, hogy az e-mail a szervezői adatok részét képezi).

**Fejléc:**

az Eventhost logót és a navigációs sávot jeleníti meg.

Bejelentkezett felhasználók számára profilképet és e-mail címet jelenít meg kijelentkezési gombbal.

Be nem jelentkezett felhasználóknak bejelentkezési gombot kínál.

Lehet, hogy tartalmaz egy hamburger menü gombot a mobilos navigációhoz.

**Test:**

Beállítja a témát "téli"-re és egy szűrőt használ a háttér elmosódításához.

Fő rész sötét háttérrel:

Megjeleníti a szervező nevét, ha az elérhető a data objektumban.

Megmutatja a szervező elérhetőségi adatait: telefonszám, ország, város, irányítószám, cím és címadatok a data objektumból.

Űrlapot jelenít meg beviteli mezőkkel az összes elérhetőségi adat szerkesztéséhez.

Rejtett beviteli mezők tárolják a data-ból lekérdezett szervezői kódot és e-mail címet. Ezeket előre kitöltik, és a felhasználó nem szerkesztheti őket.

"Mentés" gomb a frissített információk beküldéséhez ( POST kérés használatával).

## Registration

**Szkript:**

Importál szükséges elemeket és függvényeket:

A page az $app/stores-ból információkat tartalmaz az aktuális oldalról, beleértve a munkamenet adatokat (amelyek feltételezhetően felhasználói információkat tartalmaznak).

goto függvény az $app/navigation-ból egy másik oldalra való átirányításhoz.

enhance függvény az $app/forms-ból az űrlap validálásához és kezeléséhez (esetleg).

Meghatározza az data változót az oldaltartalom tárolására (a megadott kódrészletben nem használják).

A redirectTo függvény a gombra kattintáskor a kezdőoldalra irányítja a felhasználót.

**Test:**

Telepi a hátteret téli témával (szürkeárnyalatos) és elmosódási effektussal.

Fő rész sötét háttérrel:

Megjeleníti a "Finalize Your Registration" (Fejezd be a regisztrációt) címet.

Űrlapot jelenít meg beviteli mezőkkel a felhasználói adatokhoz:

Vezetéknév és keresztnév (szerkeszthető szövegmezők).

E-mail cím (csak olvasható, előre kitöltve a felhasználó e-mail címével a munkamenet adatokból).

Telefonszám, ország, város, irányítószám, cím és címadatok (mind szerkeszthető szövegmezők).

Szervezői kód (szerkeszthető szövegmező).

"Mentés" gomb, amely elküldi az űrlap adatait (POST kérés használatával) és a redirectTo függvény segítségével a kezdőoldalra irányítja a felhasználót.

## MainPage

**HTML Struktúra:**

A kód HTML elemekkel határozza meg a weboldal szerkezetét.

A <body> elem tartalmazza a weboldalon megjelenő tartalom nagy részét.

Továbbiakban különböző HTML tagekkel van al-szekciókra osztva.

Van egy fejléc szekció (<header>) a navigációhoz és a felhasználói információkhoz.

Több szekciót (<div>) használunk különálló osztályokkal, hogy különböző tartalmi területeket jelenítsünk meg egyedi stílusokkal.

**Stílusok:**

A kódolás során CSS osztályokat alkalmazunk az elemek stílusának meghatározására.

Ezek az osztályok olyan tulajdonságokat definiálnak, mint a háttér szín, szövegméret, elrendezés és reagálóképesség.

Például a snap-start osztály a görgetéskor egy elemet a nézőtér elejére pozicionál.

**Navigáció:**

A fejléc szekció (<header>) egy navigációs sávot (<nav>) tartalmaz a webhely logójával és linkekkel a különböző oldalakhoz.

A felhasználói munkamenet adatok alapján feltételesen megjeleníti a felhasználói információkat és a bejelentkezés/kijelentkezés gombot is.

**Tartalmi szekciók:**

A weboldal több, különböző tartalommal és stílussal rendelkező szekciót mutat be.

Az egyik szekció elmosódási effektusú háttérképet használ, és megjeleníti a webhely címét és leírását.

Más szekciók szöveges tartalmat tartalmaznak, amelyek a vállalat szolgáltatásait és előnyeit ismertetik.

A vizuális megjelenítés fokozása érdekében képeket is tartalmaz.

**JavaScript (Svelte):**

A <script> tag a elején egy Svelte komponenst definiál.

Importálja a szükséges függvényeket, mint például a signIn és a signOut a felhasználói hitelesítéshez (egy hitelesítési könyvtárból).

Meghatároz egy onMount életciklus-függvényt, amely a komponens betöltődésekor futtatódik.

Ez a függvény lekéri a "eyeCatcher" azonosítójú elemet, és finoman görgeti azt a nézőtérbe.

# Backend:

## AboutUs

**Importálások:**

* A RequestEvent interface-t a ./$types elérési útvonalról importáljuk (a kérelem esemény adatait definiálja).
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A PageServerLoad interface-t a ./$types elérési útvonalról importáljuk. Ez definiálja a függvény által visszaadandó adatok szerkezetét.
* A client konstans az "@/lib/server/edgedb" elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.
* A redirect függvényt az @sveltejs/kit könyvtárból importáljuk az átirányítás kezeléséhez.

load **Funkció:**

* Ez a függvény a PageServerLoad interface-t implementálja, ami a szerveroldali adatbeolvasást jelenti.
* A ({ locals }) argumentumot fogadja, mely a kérelem esemény adatait tartalmazza (RequestEvent).
* Lekéri a felhasználói munkamenetet a locals.auth() függvénnyel.
* Kinyeri a felhasználó e-mail címét a munkamenetből, ha be van jelentkezve.
* Két EdgeQL lekérdezést hajt végre párhuzamosan ( .run(client)):
  + Az első lekérdezi a felhasználókat az e-mail cím alapján.
  + A második lekérdezi a szervezőket az e-mail cím alapján.
* A lekérdezések eredményeit tömbökbe rendezi.
* A felhasználói e-mail cím alapján megkeresi a felhasználót és a szervezőt a lekérdezés eredményei között.
* Egy újabb logikai változó (emailFound) tárolja, hogy találtak-e e-mail címet a felhasználók vagy a szervezők között.

**Felhasználó Átirányítás (Nincs E-mail Találat):**

* Ha a felhasználó be van jelentkezve (session?.user), de az e-mail címe nem található sem a felhasználók, sem a szervezők között (!emailFound), a kód 307-es HTTP átirányítást hajt végre a regisztrációs oldalra (/registration). Ez arra utalhat, hogy a felhasználónak frissítenie kell a regisztrációját, vagy az e-mail címe nincs megfelelően az adatbázisban.

**Szerepkör Alapú Adatvisszatérés (Bejelentkezett Felhasználó):**

* Ha a felhasználó e-mail címe megtalálható a szervezők között (foundOrganizerEmail), a kód a szervező adatait tartalmazó objektumot ad vissza egy újabb EdgeQL lekérdezés segítségével.
* Ha a felhasználó e-mail címe megtalálható a felhasználók között (foundUserEmail), a kód a felhasználó adatait tartalmazó objektumot ad vissza egy újabb EdgeQL lekérdezés segítségével.

## Company

1. **Importálások:**
   * A PageServerLoad interface-t a ./$types elérési útvonalról importáljuk. Ez a függvény által visszaadott adatok szerkezetét definiálja.
   * Az error függvényt az @sveltejs/kit könyvtárból importáljuk a hibakezeléshez.
   * Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
   * A client konstans az '$lib/server/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.
2. load **Funkció:**
   * Ez a függvény PageServerLoad típusú és exportált, ami azt jelenti, hogy a szerveroldali adatbeolvasásért felelős.
   * A ({ params, locals }) argumentumokat fogadja:
     + A params objektum az URL elérési útvonal paramétereit tartalmazza.
     + A locals objektum az előző kérelemkezelőktől származó információkat tartalmazza (pl. felhasználói hitelesítés adatai).
3. **Felhasználói Munkamenet Lekérése:**
   * A const session = await locals.auth() sor a felhasználói munkamenet adatait kérdezi le az auth függvény segítségével a locals objektumból.
4. **Vállalati Adatok Lekérése:**
   * EdgeQL lekérdezést használ a kód a URL paraméterekben (params.id) megadott azonosítóval rendelkező vállalat adatainak lekérésére.
   * A lekérdezés a vállalat összes mezőjét is lekéri (...e.Company['\*']).
5. **Felhasználó E-mail Címének és Domain Nevének Kinyerése:**
   * A kód a munkamenetből kinyeri a felhasználó e-mail címét, feltételezve, hogy be van jelentkezve (session?.user?.email).
   * Ezután az e-mail címet a "@" karakter mentén vágja szét, hogy megkapja a domain nevet (afterAt).
   * A domain nevet sztringgé alakítja (afterAtString).
   * Ellenőrzi, hogy a vállalat e-mail címe tartalmazza-e a felhasználó domain nevét (emailMatched).
6. **Szervezői és Felhasználói Adatok Lekérése (Feltételes):**
   * Ha a felhasználó domain neve megegyezik a vállalat e-mail címének domain nevével:
     + A kód EdgeQL lekérdezést hajt végre egy olyan szervező adatainak lekérésére, akinek az e-mail címe megegyezik a felhasználó e-mail címével (searchSessionEmailString).
     + Hasonlóan lekérdezi azt a felhasználót is, akinek az e-mail címe megegyezik a felhasználó e-mail címével.
   * A lekérdezések csak a email és id mezőket kérik le a hatékonyság érdekében.
7. **Adatok Visszatérése vagy Hiba Dobása:**
   * Ha az emailMatched érték igaz, akkor a függvény egy objektumot ad vissza, amely a lekérdezett company (vállalat), organizer (szervező) és user (felhasználó) adatokat tartalmazza.
   * Ellenkező esetben, ha a vállalat nem található, vagy az e-mail domainek nem

## Company/[id=uuid]/update

1. **Adatok betöltése:**
   * A komponens betöltésekor lekérdezi egy vállalat adatait az URL paraméterekben megadott azonosító alapján.
   * Ellenőrzi a felhasználó jogosultságát a lekérdezett vállalathoz a felhasználó e-mail címe alapján.
   * Csak akkor ad vissza adatokat, ha a felhasználó jogosult a vállalat megtekintésére.
2. **Adatok frissítése:**
   * A komponens egy űrlapot kezel, amely a felhasználó által megadott adatokkal frissíti a vállalat adatait.
   * A kód érvényesíti a felhasználói bevitelt egy Zod sémával, mielőtt frissítené az adatbázist.
   * Sikeres frissítés esetén nem ad vissza semmit.
   * Hiba esetén (pl. érvénytelen bevitel, jogosultsági probléma) a kód hibát dob vissza.

**Importálások:**

* A PageServerLoad és Actions interface-eket a ./$types elérési útvonalról importáljuk. Ezek definiálják a függvények által visszaadandó adatok szerkezetét.
* Az error és fail függvényeket az @sveltejs/kit könyvtárból importáljuk a hibakezeléshez.
* A superValidate függvényt a sveltekit-superforms/server könyvtárból importáljuk az űrlapadatok érvényesítéséhez.
* A z konstans a Zod könyvtárból származik, melyet az űrlapadatok séma validálásához használunk.
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A client konstans az '@/services/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.

load **Funkció:**

* Ez a függvény a PageServerLoad interface-t implementálja, ami a szerveroldali adatbeolvasást jelenti.
* Az ({ params, locals }) argumentumokat fogadja:
  + A params objektum az URL elérési útvonal paramétereit tartalmazza.
  + A locals objektum a felhasználói munkamenet és egyéb kérelemkezelőktől származó információkat tartalmazhat.
* Lekéri a felhasználói munkamenetet a locals.auth() függvénnyel.
* EdgeQL lekérdezést használ a vállalat adatainak lekéréséhez a params.id paraméterben megadott azonosító alapján.
* Ellenőrzi, hogy a felhasználó e-mail címe tartalmazza-e a lekérdezett vállalat e-mail címének domain nevét.
* Csak akkor kérdezi le a szervező adatait, ha a felhasználó jogosult a vállalat megtekintésére.
* A függvény vagy a lekérdezett vállalat és szervező adatait tartalmazó objektumot ad vissza, vagy 404-es hibát dob, ha a vállalat nem található, vagy a felhasználónak nincs jogosultsága.

actions **Objektum:**

* Ez az objektum az űrlapkezelő függvényeket tartalmazza.
* A default függvény a superValidate függvénnyel érvényesíti a felhasználó által az űrlapon megadott adatokat a meghatározott Zod séma alapján.
* Érvénytelen bevitel esetén 400-as hibát dob vissza a kitöltött űrlap adatokkal.
* Érvényes bevitel esetén EdgeQL lekérdezést használ a vállalat adatainak frissítésére a megadott adatokkal.

## Events

**Importálások:**

* A PageServerLoad interface-t a ./$types elérési útvonalról importáljuk. Ez definiálja a függvény által visszaadandó adatok szerkezetét.
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A client konstans az '$lib/server/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.
* A redirect függvényt az @sveltejs/kit könyvtárból importáljuk az átirányítás kezeléséhez.
* A RequestEvent interface-t a ./$types elérési útvonalról importáljuk (a kérelem esemény adatait definiálja).

load **Funkció:**

* Ez a függvény a PageServerLoad interface-t implementálja, ami a szerveroldali adatbeolvasást jelenti.
* A ({ locals }) argumentumot fogadja, mely a kérelem esemény adatait tartalmazza (RequestEvent).
* Lekéri a felhasználói munkamenetet a locals.auth() függvénnyel.
* Kinyeri a felhasználó e-mail címét a munkamenetből, ha be van jelentkezve.
* Az e-mail cím domain nevét kinyeri az e-mail címből a "@" karakter mentén.
* EdgeQL lekérdezésekkel a felhasználóhoz tartozó szervezőt és felhasználót kérdezi le az e-mail cím alapján.

**Események Lekérése (Bejelentkezett Felhasználó):**

* Ha a felhasználó be van jelentkezve, a kód egy további EdgeQL lekérdezést hajt végre az eseményekre.
* A lekérdezés szűri az eseményeket a felhasználó e-mail címének domain neve alapján (emailValidation mező).
* A lekérdezés az esemény következő mezőit kéri le:
  + id
  + title (cím)
  + placeName (helyszín neve)
  + startsAt (kezdési időpont)
  + endsAt (befejezési időpont)
  + description (leírás)
  + createdAt (létrehozás időpontja)
  + organizer (szervező adatai: firstName és lastName)
* A lekérdezés eredménye, a szervező és a felhasználó adatai kerülnek visszaadásra.

**Átirányítás (Kijelentkezett Felhasználó):**

* Ha a felhasználó nincs bejelentkezve, a kód 307-es HTTP átirányítást hajt végre a kezdőoldalra (/).

## Events/[id=uuid]

**Importálások:**

* A PageServerLoad interface-t a ./$types elérési útvonalról importáljuk. Ez definiálja a függvény által visszaadandó adatok szerkezetét.
* Az error függvényt az @sveltejs/kit könyvtárból importáljuk a hibakezeléshez.
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A client konstans az '$lib/server/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.

load **Funkció:**

* Ez a függvény a PageServerLoad interface-t implementálja, ami a szerveroldali adatbeolvasást jelenti.
* A ({ params, locals }) argumentumokat fogadja:
  + A params objektum az URL elérési útvonal paramétereit tartalmazza.
  + A locals objektum a felhasználói munkamenetet és egyéb kérelemkezelőktől származó információkat tartalmazhat.
* Lekéri a felhasználói munkamenetet a locals.auth() függvénnyel.
* Kinyeri a felhasználó e-mail címét a munkamenetből, ha be van jelentkezve.
* Az e-mail cím domain nevét kinyeri az e-mail címből a "@" karakter mentén.
* EdgeQL lekérdezéssel lekéri az esemény adatait a params.id paraméterben megadott azonosító alapján.

**Felhasználói Jogosultság Ellenőrzése:**

* A kód ellenőrzi, hogy a felhasználó e-mail címének domain neve megegyezik-e az esemény emailValidation mezőjében tárolt domain névvel.
* Csak akkor folytatódik a kód futtatása, ha a domain nevek megegyeznek.

**További Adatok Lekérése ( Jogosult Felhasználó):**

* Ha a felhasználó jogosult az esemény megtekintésére, a kód további EdgeQL lekérdezéseket hajt végre a szervező és a felhasználó adatainak lekérése érdekében.

**Adatok Visszatérése vagy Hiba Dobása:**

* Ha a felhasználó jogosult az esemény megtekintésére, a kód egy objektumot ad vissza, amely az esemény, a szervező és a felhasználó adatait tartalmazza.
* Ha az esemény nem található, vagy a felhasználó nem jogosult a megtekintésére, a kód 404-es hibát dob.

## Events/[id=uuid]/update

1. **Adatok betöltése:**
   * A komponens betöltésekor lekéri egy esemény adatait az URL paraméterekben megadott azonosító alapján.
   * Ellenőrzi a felhasználó jogosultságát az esemény megtekintéséhez a felhasználó e-mail címe alapján.
   * Csak akkor ad vissza adatokat, ha a felhasználó jogosult az esemény megtekintésére.
2. **Adatok frissítése:**
   * A komponens egy űrlapot kezel, amely a felhasználó által megadott adatokkal frissíti az esemény adatait.
   * A kód érvényesíti a felhasználói bevitelt egy Zod sémával, mielőtt frissítené az adatbázist.
   * Érvénytelen bevitel esetén (pl. érvénytelen dátum, jogosultsági probléma) a kód hibát dob vissza.

**Importálások:**

* A PageServerLoad és Actions interface-eket a ./$types elérési útvonalról importáljuk. Ezek definiálják a függvények által visszaadandó adatok szerkezetét.
* A fail függvényt az @sveltejs/kit könyvtárból importáljuk a hibakezeléshez.
* A superValidate függvényt a sveltekit-superforms/server könyvtárból importáljuk az űrlapadatok érvényesítéséhez.
* A z konstans a Zod könyvtárból származik, melyet az űrlapadatok séma validálásához használunk.
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A client konstans az '@/services/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.
* Az error függvényt a console modulból importáljuk a konzolos hibakereséshez (de a kód inkább a throw error(404) hibadobást használja).

load **Funkció:**

* Ez a függvény a PageServerLoad interface-t implementálja, ami a szerveroldali adatbeolvasást jelenti.
* A ({ params, locals }) argumentumokat fogadja:
  + A params objektum az URL elérési útvonal paramétereit tartalmazza.
  + A locals objektum a felhasználói munkamenetet és egyéb kérelemkezelőktől származó információkat tartalmazhat.
* Lekéri a felhasználói munkamenetet a locals.auth() függvénnyel.
* Kinyeri a felhasználó e-mail címét a munkamenetből, ha be van jelentkezve.
* EdgeQL lekérdezéssel lekéri a felhasználóhoz tartozó szervezőt az e-mail cím alapján.
* EdgeQL lekérdezéssel lekéri az esemény adatait a params.id paraméterben megadott azonosító alapján.
* Ellenőrzi, hogy a felhasználó e-mail címének domain neve megegyezik-e az esemény emailValidation mezőjében tárolt domain névvel.
* Csak akkor ad vissza adatokat (esemény és szervező), ha a felhasználó jogosult az esemény megtekintésére (e-mail cím domain neve egyezik, és a felhasználó szervezője az esemény szervezője).

actions **Objektum:**

* Ez az objektum az űrlapkezelő függvényt tartalmazza.
* A default függvény a superValidate függvénnyel érvényesíti a felhasználó által az űrlapon megadott adatokat a meghatározott Zod sémával. A séma többek között ellenőrzi, hogy az esemény kezdő dátuma megelőzze a befejező dátumot.
* Érvénytelen bevitel esetén 400-as hibát dob vissza a kitöltött űr

## Home

**Importálások:**

* A redirect függvényt az @sveltejs/kit könyvtárból importáljuk az átirányítás kezeléséhez.
* A RequestEvent interface-t a ./$types elérési útvonalról importáljuk ( a kérelem esemény adatait definiálja).
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A PageServerLoad interface-t a ./$types elérési útvonalról importáljuk. Ez definiálja a függvény által visszaadandó adatok szerkezetét.
* A client konstans az "@/lib/server/edgedb" elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.

load **Funkció:**

* Ez a függvény a PageServerLoad interface-t implementálja, ami a szerveroldali adatbeolvasást jelenti.
* A ({ locals }) argumentumot fogadja, mely a kérelem esemény adatait tartalmazza (RequestEvent).
* Lekéri a felhasználói munkamenetet a locals.auth() függvénnyel.
* Kinyeri a felhasználó e-mail címét a munkamenetből, ha be van jelentkezve.
* Két EdgeQL lekérdezést hajt végre párhuzamosan ( .run(client)):
  + Az első lekérdezi a felhasználókat az e-mail cím alapján.
  + A második lekérdezi a szervezőket az e-mail cím alapján.
* A lekérdezések eredményeit tömbökbe rendezi.
* A felhasználói e-mail cím alapján megkeresi a felhasználót és a szervezőt a lekérdezés eredményei között.

**Felhasználó Átirányítás (Nincs Bejelentkezés):**

* Ha a felhasználó nincs bejelentkezve (!session?.user), a kód 307-es HTTP átirányítást hajt végre a kezdőoldalra (/).

**Szerepkör Alapú Adatvisszatérés (Bejelentkezett Felhasználó):**

* Ha a felhasználó be van jelentkezve:
  + Ha a felhasználó e-mail címe megtalálható a szervezők között (foundOrganizerEmail), a kód a szervező adatait tartalmazó objektumot ad vissza egy újabb EdgeQL lekérdezés segítségével.
  + Ha a felhasználó e-mail címe megtalálható a felhasználók között (foundUserEmail), a kód a felhasználó adatait tartalmazó objektumot ad vissza egy újabb EdgeQL lekérdezés segítségével.
* A kód nem tér vissza adattal, ha a bejelentkezett felhasználó sem szervező, sem normál felhasználóként nem található az adatbázisban (ez egy lehetséges hibaszcenárió lehet).

## Organizers/[id=uuid]/events/create

1. **Adatok betöltése:**
   * A komponens ellenőrzi a felhasználó jogosultságát az oldal megtekintéséhez a felhasználó e-mail címe alapján.
   * Csak akkor ad vissza adatokat (a szervező adatait), ha a felhasználó jogosult az oldal megtekintésére (a bejelentkezett felhasználó egy szervező, és az URL paraméterben megadott azonosító a szervezőhöz tartozik).
   * Jogosultsági probléma esetén a kód 307-es HTTP átirányítást hajt végre az események listájára (/events).
2. **Esemény létrehozása:**
   * A komponens egy űrlapot kezel, amely a felhasználó által megadott adatokkal hoz létre egy új eseményt.
   * A kód érvényesíti a felhasználói bevitelt egy Zod sémával, mielőtt létrehozná az eseményt az adatbázisban.
   * Érvénytelen bevitel esetén (pl. érvénytelen dátum, jogosultsági probléma) a kód hibát dob vissza.

**Importálások:**

* A PageServerLoad és Actions interface-eket a ./$types elérési útvonalról importáljuk. Ezek definiálják a függvények által visszaadandó adatok szerkezetét.
* A fail, redirect függvényeket az @sveltejs/kit könyvtárból importáljuk a hibakezeléshez és átirányításhoz.
* A superValidate függvényt a sveltekit-superforms/server könyvtárból importáljuk az űrlapadatok érvényesítéséhez.
* A z konstans a Zod könyvtárból származik, melyet az űrlapadatok séma validálásához használunk.
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A client konstans az '@/services/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.

load **Funkció:**

* Ez a függvény a PageServerLoad interface-t implementálja, ami a szerveroldali adatbeolvasást jelenti.
* A ({ locals, params }) argumentumokat fogadja:
  + A locals objektum a felhasználói munkamenetet és egyéb kérelemkezelőktől származó információkat tartalmazhat.
  + A params objektum az URL elérési útvonal paramétereit tartalmazza.
* Lekéri a felhasználói munkamenetet a locals.auth() függvénnyel.
* Kinyeri a felhasználó e-mail címét a munkamenetből, ha be van jelentkezve.
* EdgeQL lekérdezéssel lekéri a felhasználóhoz tartozó szervezőt az e-mail cím alapján.
* Összehasonlítja az URL paraméterben megadott azonosítót a szervező azonosítójával.

**Felhasználó Átirányítás (Jogosultsági Probléma):**

* Ha a felhasználó nincs bejelentkezve szervezőként (vagy a megadott azonosító nem tartozik a felhasználó szervezőjéhez), a kód 307-es HTTP átirányítást hajt végre az események listájára (/events).

**Szerepkör Alapú Adatvisszatérés ( Jogosult Szervező):**

* Ha a felhasználó egy jogosult szervező, a kód a szervező adatait tartalmazó objektumot ad vissza.

actions **Objektum:**

* Ez az objektum az űrlapkezelő függvényt tartalmazza.
* A default függvény a superValidate függvénnyel érvényesíti a felhasználó által az űrlapon megadott adatokat a meghatározott Zod sémával. A séma többek között ellenőrzi, hogy az es

## Profile/organizer/[id=uuid]

**Importálások:**

* A PageServerLoad interface-t a ./$types elérési útvonalról importáljuk. Ez definiálja a függvény által visszaadandó adatok szerkezetét.
* Az error függvényt az @sveltejs/kit könyvtárból importáljuk a hibakezeléshez.
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A client konstans az '$lib/server/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.

load **Funkció:**

* Ez a függvény a PageServerLoad interface-t implementálja, ami a szerveroldali adatbeolvasást jelenti.
* A ({ params, locals }) argumentumokat fogadja:
  + A params objektum az URL elérési útvonal paramétereit tartalmazza (ebben az esetben a szervező azonosítóját).
  + A locals objektum a felhasználói munkamenetet és egyéb kérelemkezelőktől származó információkat tartalmazhat.
* Lekéri a felhasználói munkamenetet a locals.auth() függvénnyel.
* EdgeQL lekérdezéssel lekéri a szervező adatait az URL paraméterben megadott azonosító alapján. A lekérdezés az összes mezőt lekéri (...e.Organizer['\*']).
* Kinyeri a felhasználó e-mail címét a munkamenetből, ha be van jelentkezve.
* Összehasonlítja a szervező e-mail címét a felhasználó e-mail címével.

**Felhasználó Átirányítás (Jogosultsági Probléma):**

* Ha a felhasználó e-mail címe nem egyezik meg a szervező e-mail címével, a kód 404-es hibát dob (throw error(404)) ami a legtöbb esetben az "Oldal nem található" üzenethez vezet a böngészőben. Ez arra utal, hogy a felhasználó nem jogosult a szervező adatainak megtekintésére.

**Adatvisszatérés (Jogosult Felhasználó):**

* Ha a felhasználó e-mail címe megegyezik a szervező e-mail címével, a kód a szervező adatait tartalmazó objektumot adja vissza. Ez az adat felhasználható a komponens frontendjén történő megjelenítéshez.

## Profile/organizer/[id=uuid]/update

1. **Adatok betöltése:**
   * A komponens ellenőrzi a felhasználó jogosultságát az oldal megtekintéséhez a felhasználó e-mail címe alapján.
   * Csak akkor ad vissza adatokat (a szervező adatait), ha a felhasználó jogosult az oldal megtekintésére (a bejelentkezett felhasználó egy szervező, és az URL paraméterben megadott azonosító a szervezőhöz tartozik).
   * Jogosultsági probléma esetén a kód 404-es hibát dob, ami a legtöbb esetben az "Oldal nem található" üzenethez vezet a böngészőben.
2. **Szervező adatok módosítása:**
   * A komponens egy űrlapot kezel, amely a felhasználó által megadott adatokkal frissíti a szervező adatait az adatbázisban.
   * A kód érvényesíti a felhasználói bevitelt egy Zod sémával, mielőtt módosítja a szervező adatait.
   * Érvénytelen bevitel esetén (pl. hiányzó mező, jogosultsági probléma) a kód hibát dob vissza.

**Importálások:**

* A PageServerLoad és Actions interface-eket a ./$types elérési útvonalról importáljuk. Ezek definiálják a függvények által visszaadandó adatok szerkezetét.
* Az error, fail függvényeket az @sveltejs/kit könyvtárból importáljuk a hibakezeléshez.
* A superValidate függvényt a sveltekit-superforms/server könyvtárból importáljuk az űrlapadatok érvényesítéséhez.
* A z konstans a Zod könyvtárból származik, melyet az űrlapadatok séma validálásához használunk.
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A client konstans az '@/services/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.

load **Funkció:**

* Ez a függvény a PageServerLoad interface-t implementálja, ami a szerveroldali adatbeolvasást jelenti.
* A ({ params, locals }) argumentumokat fogadja:
  + A params objektum az URL elérési útvonal paramétereit tartalmazza (ebben az esetben a szervező azonosítóját).
  + A locals objektum a felhasználói munkamenetet és egyéb kérelemkezelőktől származó információkat tartalmazhat.
* Lekéri a felhasználói munkamenetet a locals.auth() függvénnyel.
* Kinyeri a felhasználó e-mail címét a munkamenetből, ha be van jelentkezve.
* EdgeQL lekérdezéssel lekéri a szervező adatait az URL paraméterben megadott azonosító alapján. A lekérdezés az összes mezőt lekéri (...e.Organizer['\*']).
* Összehasonlítja a szervező e-mail címét a felhasználó e-mail címével.

**Felhasználó Átirányítás (Jogosultsági Probléma):**

* Ha a felhasználó e-mail címe nem egyezik meg a szervező e-mail címével, a kód 404-es hibát dob (throw error(404)) ami a legtöbb esetben az "Oldal nem található" üzenethez vezet a böngészőben. Ez arra utal, hogy a felhasználó nem jogosult a szervező adatainak megtekintésére.

**Adatvisszatérés (Jogosult Felhasználó):**

* Ha a felhasználó e-mail címe megegyezik a szervező e-mail címével, a kód a szervező adatait tartalmazó objektumot adja vissza. Ez az adat felhasználható az űr

## Profile/user/[id=uuid]

**A kód célja**

Ez a kód egy SvelteKit szerveroldali komponenst definiál, amely egy felhasználó adatait tölti be az adatbázisból az URL paraméterben megadott azonosító alapján. A kód ellenőrzi a felhasználó jogosultságát az adatok megtekintéséhez a felhasználó e-mail címe alapján.

**Importálások:**

* A PageServerLoad interface-t a ./$types elérési útvonalról importáljuk. Ez definiálja a függvény által visszaadandó adatok szerkezetét.
* Az error függvényt az @sveltejs/kit könyvtárból importáljuk a hibakezeléshez.
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A client konstans az '$lib/server/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.

load **Funkció:**

* Ez a függvény a PageServerLoad interface-t implementálja, ami a szerveroldali adatbeolvasást jelenti.
* A ({ params, locals }) argumentumokat fogadja:
  + A params objektum az URL elérési útvonal paramétereit tartalmazza (ebben az esetben a felhasználó azonosítóját).
  + A locals objektum a felhasználói munkamenetet és egyéb kérelemkezelőktől származó információkat tartalmazhat.
* Lekéri a felhasználói munkamenetet a locals.auth() függvénnyel.
* Kinyeri a felhasználó e-mail címét a munkamenetből, ha be van jelentkezve.
* EdgeQL lekérdezéssel lekéri a felhasználó adatait az URL paraméterben megadott azonosító alapján. A lekérdezés az összes mezőt lekéri (...e.User['\*']).
* Összehasonlítja a felhasználó e-mail címét a lekérdezett felhasználó e-mail címével.

**Felhasználó Átirányítás (Jogosultsági Probléma):**

* Ha a felhasználó e-mail címe nem egyezik meg a lekérdezett felhasználó e-mail címével, a kód 404-es hibát dob (throw error(404)) ami a legtöbb esetben az "Oldal nem található" üzenethez vezet a böngészőben. Ez arra utal, hogy a felhasználó nem jogosult a másik felhasználó adatainak megtekintésére.

**Adatvisszatérés (Jogosult Felhasználó):**

* Ha a felhasználó e-mail címe megegyezik a lekérdezett felhasználó e-mail címével, a kód a felhasználó adatait tartalmazó objektumot adja vissza. Ez az adat felhasználható a komponens frontendjén történő megjelenítéshez.

## Profile/user/[id=uuid]/update

**1. Adatok betöltése (**load **függvény):**

* A load függvény a felhasználó adatait tölti be az adatbázisból az URL paraméterben megadott azonosító alapján.
* Ellenőrzi a felhasználó jogosultságát az adatok megtekintéséhez a bejelentkezett felhasználó e-mail címe alapján.
* Csak akkor ad vissza adatokat (a felhasználó adatait), ha a felhasználó jogosult az oldal megtekintésére (a bejelentkezett felhasználó e-mail címe megegyezik a lekérdezett felhasználó e-mail címével).
* Jogosultsági probléma esetén 404-es HTTP hibát dob, ami a legtöbb esetben az "Oldal nem található" üzenethez vezet a böngészőben.

**2. Felhasználói adatok módosítása (**actions **objektum):**

* Az actions objektum default tulajdonsága egy olyan async függvényt tartalmaz, amely az űrlapadatokat kezeli.
* A felhasználó által az űrlapon megadott adatokat a Zod könyvtár séma segítségével validálja.
* Érvénytelen bevitel esetén (pl. hiányzó mező, hibás formátum) 400-as HTTP hibát ad vissza a fail függvénnyel.
* Érvényes bevitel esetén EdgeQL lekérdezést használ a felhasználó adatainak frissítéséhez az adatbázisban. A lekérdezés az URL paraméterben megadott azonosító alapján szűri a felhasználót, majd a megadott mezőket frissíti az űrlapadatok alapján.

**Importálások:**

* A PageServerLoad és Actions interface-eket a ./$types elérési útvonalról importáljuk. Ezek definiálják a függvények által visszaadandó adatok szerkezetét.
* Az error, fail függvényeket az @sveltejs/kit könyvtárból importáljuk a hibakezeléshez.
* A superValidate függvényt a sveltekit-superforms/server könyvtárból importáljuk az űrlapadatok validálásához.
* A z konstans a Zod könyvtárból származik, melyet az űrlapadatok séma validálásához használjuk.
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A client konstans az '$lib/server/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.

**Séma (Űrlapadatok validálás):**

* A schema konstans egy Zod sémát definiál, amely meghatározza az űrlapadatok típusát és validálási szabályait.
* A séma olyan mezőket tartalmaz, mint ország, irányítószám, város, cím, e-mail cím, keresztnév, vezetéknév, telefon és szervezői kód.
* A Zod segítségével ellenőrizhetjük, hogy a mezők értékei a megfelelő típusúak és megfelelnek az esetlegesen megadott validálási szabályoknak (pl. kötelező mező, minimális hosszúság).

## Registration

Ez a kód egy SvelteKit szerveroldali komponenst definiál, amely a felhasználó regisztrációját kezeli. A komponens két fő funkciót valósít meg:

**1.Hozzáférés-ellenőrzés (**load **függvény):**

* A load függvény ellenőrzi, hogy a felhasználó már be van-e jelentkezve.
* Ha a felhasználó be van jelentkezve, átirányítja a felhasználót a főoldalra (/) 307-es HTTP átirányítással. A 307-es átirányítás azt jelenti, hogy a kérés továbbra is a POST metódust használja az új oldalon.

**2. Felhasználói regisztráció (**actions **objektum):**

* Az actions objektum default tulajdonsága egy olyan async függvényt tartalmaz, amely kezeli a regisztrációs űrlap űrlapadatokat.
* A felhasználó által az űrlapon megadott adatokat a Zod könyvtár séma segítségével validálja.
* Érvénytelen bevitel esetén (pl. hiányzó mező, hibás formátum) 400-as HTTP hibát ad vissza a fail függvénnyel.
* Érvényes bevitel esetén a következőket ellenőrzi:
  + Van-e már regisztrált szervezet a megadott szervezeti kóddal (organizerCode).
  + A megadott e-mail cím domain része megegyik-e a valamelyik regisztrált szervezet e-mail címének domain részével.
* Ha a szervezeti kód érvénytelen vagy az e-mail cím domain része nem egyezik, 400-as HTTP hibát ad vissza a fail függvénnyel.
* Ha a szervezeti kód érvényes és az e-mail cím domain része egyezik, akkor a megadott adatok alapján a következőket hoz létre:
  + Új szervezet, ha még nincs ilyen szervezet az adatbázisban.
  + Új felhasználó, ha a felhasználó még nincs regisztrálva az adott domainhez tartozó szervezettel.

**Importálások:**

* A PageServerLoad és Actions interface-eket a ./$types elérési útvonalról importáljuk. Ezek definiálják a függvények által visszaadandó adatok szerkezetét.
* A fail, redirect függvényeket az @sveltejs/kit könyvtárból importáljuk a hibakezeléshez és átirányításhoz.
* A superValidate függvényt a sveltekit-superforms/server könyvtárból importáljuk az űrlapadatok validálásához.
* A e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A client konstans az '$lib/server/edgedb' elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.
* A z konstans a Zod könyvtárból származik, melyet az űrlapadatok séma validálásához használjuk.

**Séma (Űrlapadatok validálás):**

* A schema konstans egy Zod sémát definiál, amely meghatározza az űrlapadatok típusát és validálási szabályait.
* A séma olyan mezőket tartalmaz, mint cím, címp補足 (cím kiegészítés), város, ország, irányítószám, e-mail cím, keresztnév, vezetéknév, telefon és szervezői kód.
* A Zod segítségével ellenőrizhetjük, hogy a mezők értékei a megfelelő típusúak és legalább egy karaktert tartalmaznak.

## MainPage

**1. Felhasználói adatok lekérése:**

* A load függvény ellenőrzi, hogy a felhasználó be van-e jelentkezve.
* Ha a felhasználó be van jelentkezve, lekéri a felhasználó e-mail címét a munkamenetből.
* EdgeQL lekérdezésekkel megpróbálja megtalálni a felhasználót e-mail cím alapján a felhasználók (e.User) és a szervezetek (e.Organizer) táblákban.

**2. Átirányítás a regisztrációs oldalra (Hozzáférés-ellenőrzés):**

* Ha a bejelentkezett felhasználó e-mail címe nem található sem a felhasználók, sem a szervezetek között, akkor a komponens 307-es HTTP átirányítást hajt végre a /registration oldalra. A 307-es átirányítás azt jelenti, hogy a kérés továbbra is a POST metódust használja az új oldalon.

**3. Felhasználói adatok visszaadása:**

* Ha a felhasználó e-mail címe megtalálható a felhasználók táblában (foundUserEmail igaz), a komponens visszaad egy objektumot, amely a felhasználó adatait tartalmazza (user).
* Ha a felhasználó e-mail címe megtalálható a szervezetek táblában (foundOrganizerEmail igaz), a komponens visszaad egy objektumot, amely a szervezet adatait tartalmazza (organizer).

**Importálások:**

* A redirect függvényt az @sveltejs/kit könyvtárból importáljuk az átirányításhoz.
* A RequestEvent típust a ./$types elérési útvonalról importáljuk.
* Az e aliast az '@/edgeql-js' elérési útvonalról importáljuk, ami egy segítőkönyvtár az EdgeQL lekérdező nyelv használatához az adatbázis eléréséhez.
* A PageServerLoad típust a ./$types elérési útvonalról importáljuk.
* A client konstans az "@/lib/server/edgedb" elérési útvonalról importált adatbázis kapcsolat az EdgeQL adatbázishoz.

# Adatbázis:

**Típusok:**

* **HasAddress (Címmel rendelkezik):** Ez egy absztrakt típus, amely a címmel rendelkező entitások közös tulajdonságait határozza meg. Tartalmaz kötelező mezőket az "ország", "irányítószám", "város", "cím" és "címrészletek" számára.
* **HasTimestamps (Időbélyeggel rendelkezik):** Egy másik absztrakt típus, amely az időbélyeggel rendelkező entitások tulajdonságait határozza meg. Tartalmazza a createdAt (Létrehozva) és az updatedAt (Frissítve) mezőket, amelyek mindkettő dátumidő. A createdAt automatikusan beállítódik az entitás létrehozásának aktuális időpontjára, míg az updatedAt minden módosításkor frissül.
* **Felhasználó:** Ez a típus a HasAddress és a HasTimestamps típust is kiterjeszti. A rendszer egy felhasználóját reprezentálja, és tartalmaz olyan további mezőket, mint a keresztnév, vezetéknév, email (egyedi korlátozás), telefon és egy opcionális szervezőkód.
* **Esemény:** Ez a típus szintén a HasAddress és a HasTimestamps típust kiterjeszti. Egy eseményt reprezentál, és tartalmaz olyan részleteket, mint a cím, helyszín, emailValidálás, opcionális leírás, kezdés és befejezés időbélyegek (korlátozással, hogy az esemény a befejezés előtt kezdődjön), kezdésÓra egy meghatározott formátumban, kötelező hivatkozás a Szervezőhöz, és egy többhivatkozás a kapcsolódó Jegy entitásokra.
* **Szervező:** A Felhasználóhoz hasonlóan ez a típus is a HasAddress és a HasTimestamps típust kiterjeszti, és egy eseményszervezőt reprezentál. Tartalmazza a keresztnév, vezetéknév, egyedi email, telefon, kötelező szervezőkód mezőket, valamint egyetlen hivatkozást a szervezett Eseményre.
* **Cég:** Ez a típus a HasAddress és a HasTimestamps típust kiterjeszti, és egy céget reprezentál. Olyan részleteket tartalmaz, mint a név, egyedi vállalatiEmail, adószám, szervezőkód, valamint opcionális mezőket a webhelyhez, leíráshoz, közösségi média linkekhez és szálakhoz.
* **cegHir (Vállalati hírek):** Ez a típus csak a HasTimestamps típust kiterjeszti, és a vállalati híreket reprezentálja. Opcionális mezői vannak a cím és a leírás számára.
* **Jegy:** Ez a típus a HasTimestamps típust kiterjeszti, és egy eseményre szóló jegyet reprezentál. Tartalmaz egyedi tokent, a résztvevő adatait, mint a teljes név, keresztnév, vezetéknév, email, telefon és meghívó kód, opcionális szervezetNeve mezőt, valamint kötelező hivatkozást a kapcsolódó Eseményhez.

**Korlátozások:**

* Több mezőre is vonatkoznak korlátozások:
  + A Felhasználó és a Szervező email mezője egyedi (kizárólagos).
  + Az Esemény kezdés és befejezés mezői biztosítják, hogy az esemény a befejezés előtt kezdődjön.
  + Az Esemény kezdésÓra mezője reguláris kifejezés korlátozással rendelkezik a formátum (HH:MM) érvényesítéséhez.
  + A Jegy token mezője szintén egyedi (kizárólagos).

**Kapcsolatok:**

**Esemény:** Kötelező hivatkozással rendelkezik egy **Szervezőre**.

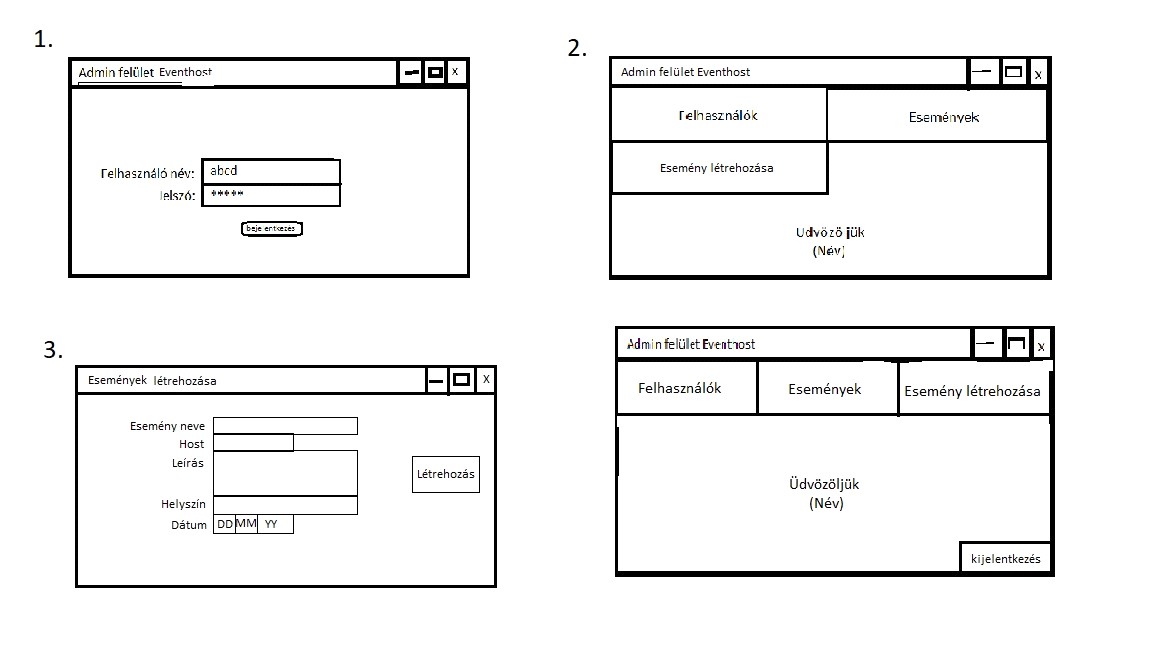
**Szervező:** Egyetlen hivatkozással rendelkezik a szervezett **Eseményre**.

**Esemény:** Többhivatkozással rendelkezik a kapcsolódó **Jegy** entitásokra.

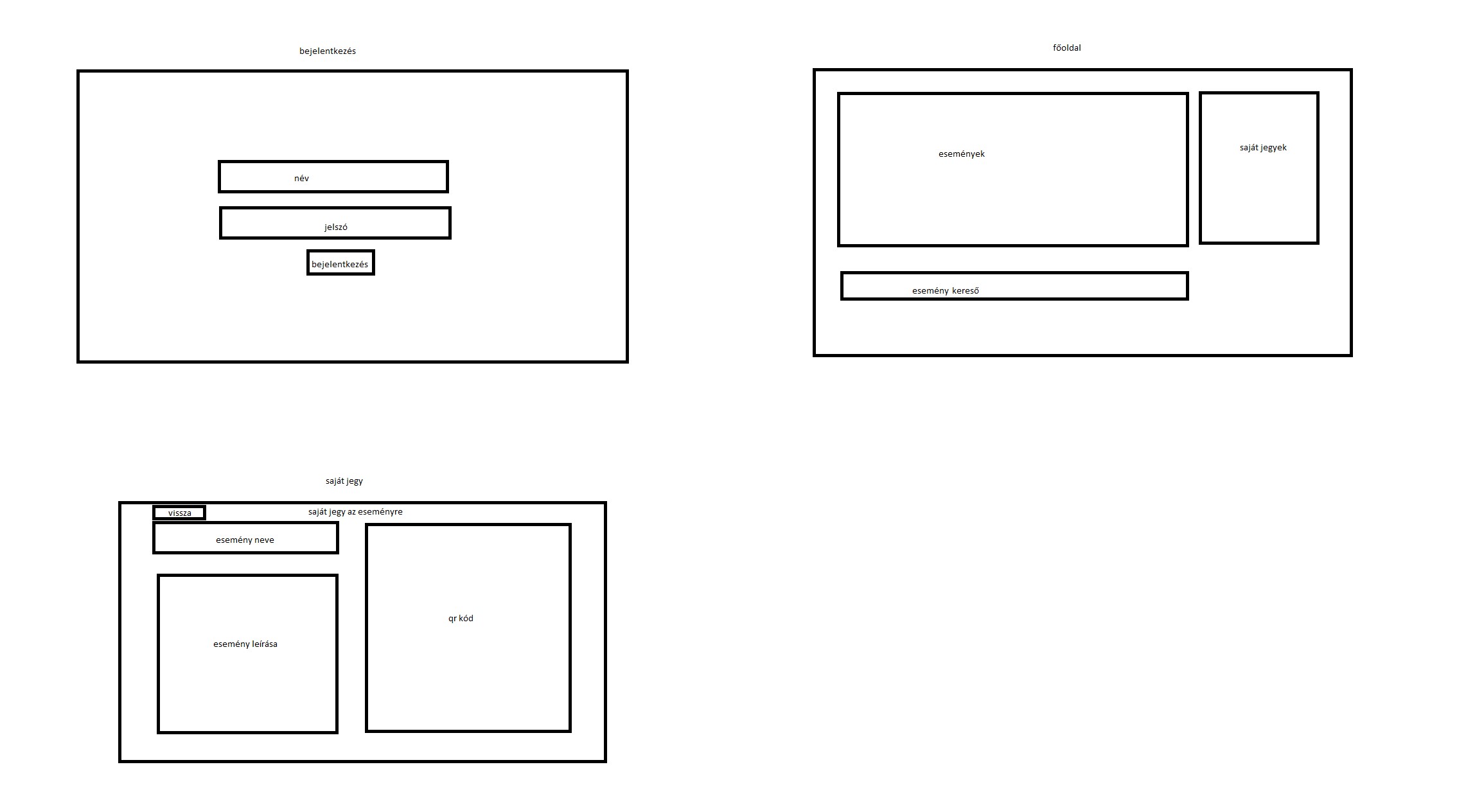
**Jegy:** Kötelező hivatkozással rendelkezik a kapcsolódó **Eseményhez**.

# Tervek:

Asztali alkalmazás terv:



Kinézet terv:



Adatbázis:

