Családom Emléke

Tartalom

[**Vízió:** 2](#_Toc133234845)

[**Fejlesztői környezet:** 2](#_Toc133234846)

[**Követelmények:** 2](#_Toc133234847)

[**Futtatáshoz szükséges szoftverek:** 2](#_Toc133234848)

[**Szükséges node modulok:** 2](#_Toc133234849)

[**Inicializálás:** 3](#_Toc133234850)

[**Futtatás:** 3](#_Toc133234851)

[**Mappaszerkezet:** 3](#_Toc133234852)

[**Backend:** 3](#_Toc133234853)

[**Frontend:** 4](#_Toc133234854)

[**Fájlok:** 4](#_Toc133234855)

[**Adatbázis:** 5](#_Toc133234856)

[**Kommunikáció:** 6](#_Toc133234857)

[**Authorizáció (token):** 6](#_Toc133234858)

[**Kimenet:** 6](#_Toc133234859)

[**Tesztek:** 6](#_Toc133234860)

**Vízió:**

A felhasználónak lehetősége van rögzíteni, szerkeszteni és törölni saját vagy fikciós családfáját. A benne szereplő családtagokhoz feltölthet paragrafusokat életútjuk gyanánt, valamint képeket adhat hozzá az adott személyhez. Amennyiben a családfa publikus QR kódot generálva a linkről elérhetőséget készíthet és oszthat meg közösségi oldalain vagy akár tölthet le kép formájában. Ezekről a linkekről a családfa személyei és ezáltal a családfa megtekinthető bejelentkezés nélkül is amennyiben publikus státusszal rendelkezik.

**Fejlesztői környezet:**

Operációs rendszer: Microsoft Windows 10

IDE: Visual Studio Code

Böngészők:

* Firefox,
* Google Chrome,
* Microsoft Edge
* Brave

Backend: NodeJS 18.13.0

Frontend: VueJS 3.2.47

Adatbázis: MySQL

Adatbázis futtató alkalmazás: XAMPP v3.2.4

**Követelmények:**

**Futtatáshoz szükséges szoftverek:**

* node (16.13.0, vagy újabb)
* npm (8.19.3 vagy újabb)
* XAMPP (3.2.4 vagy újabb)

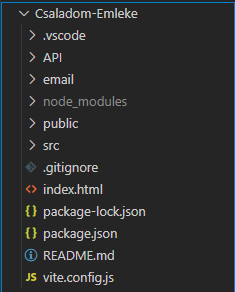
**Szükséges node modulok:**

* dotenv (16.0.3)
* express (4.18.2)
* cors (2.8.5)
* path (0.12.7)
* moment (2.29.4)
* mysql (2.18.1)
* multer (1.14.5)
* fs
* jsonwebtoken (9.0.0)
* qrcode (1.5.1)
* nodemailer (6.9.1)
* ejs (3.1.9)

**Inicializálás:**

* A „**Csaladom-Emleke**” mappában az „**npm i**” paranccsal telepítjük a frontend futtatásához szükséges node modulokat.
* A „**Csaladom-Emleke/API**” mappában az „**npm i**” paranccsal telepítjük a backend futtatásához szükséges node modulokat.
* Az „**Csaladom-Emleke/API/sql**” mappa **sql** állományát importáljuk az adatbázisunkba.

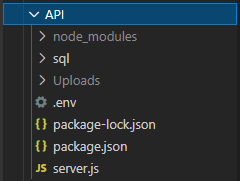
**Futtatás:**

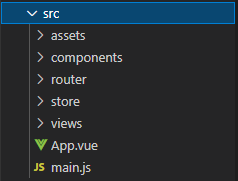
* A „**Csaladom-Emleke**” mappában az „**npm run dev**” paranccsal futtatjuk a **frontend** részét az alkalmazásnak
* A „**Csaladom-Emleke/API**” mappában az „**npm run dev**” paranccsal futtatjuk a **backend** részét az alkalmazásnak

**Mappaszerkezet:**

* API mappa: tartalmazza a backend fájljait.
* email mappa: tartalmazza az elfelejtett jelszó  
  funkció által küldött e-mail sablont. (ejs)
* public mappa: tartalmazza a favicont
* src mappa: tartalmazza a frontend fájljait

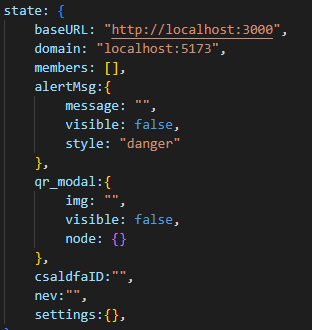
**Backend:**

* sql mappa: tartalmazza az adatbázis sql fájlját
* Uploads mappa: a családtagok képeit illetve  
  QR kódjait tartalmazza PNG formátumban

**Frontend:**

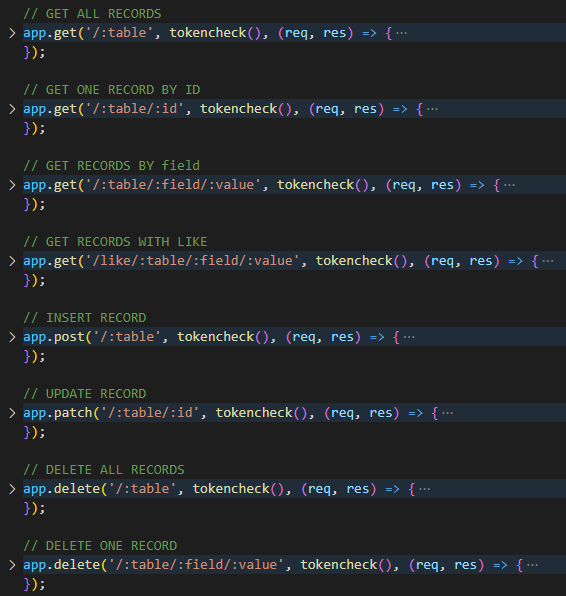
* assets mappa: a weboldalon statikusan használt képek
* components mappa: frontend oldalon felhasznált  
  komponensek
* router mappa: vue router
* store mappa: vue store amely a globális változókat tartalmazza
* views: a router által kiválasztott nézeteket tartalmazza

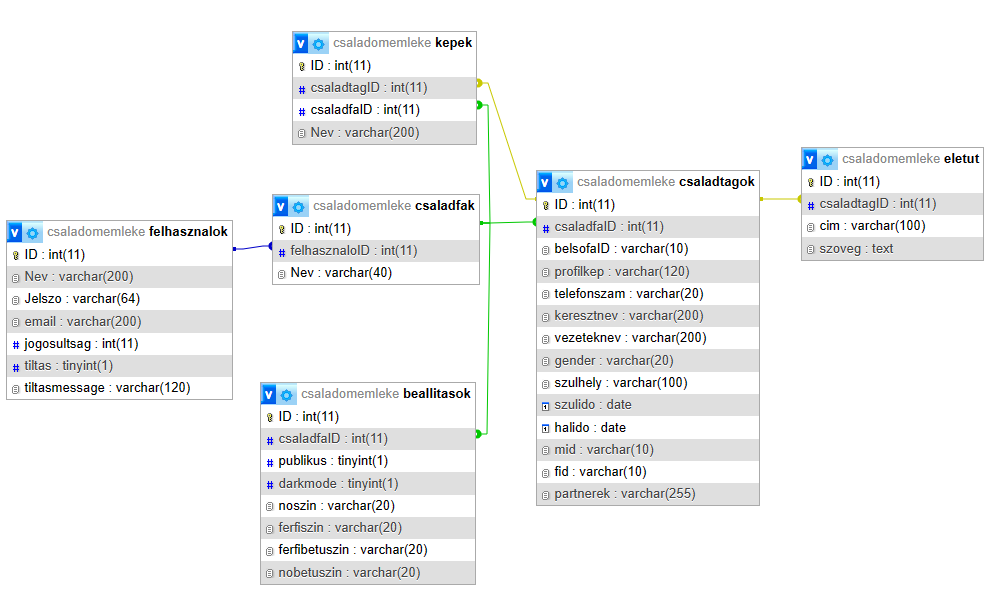
**Fájlok:**

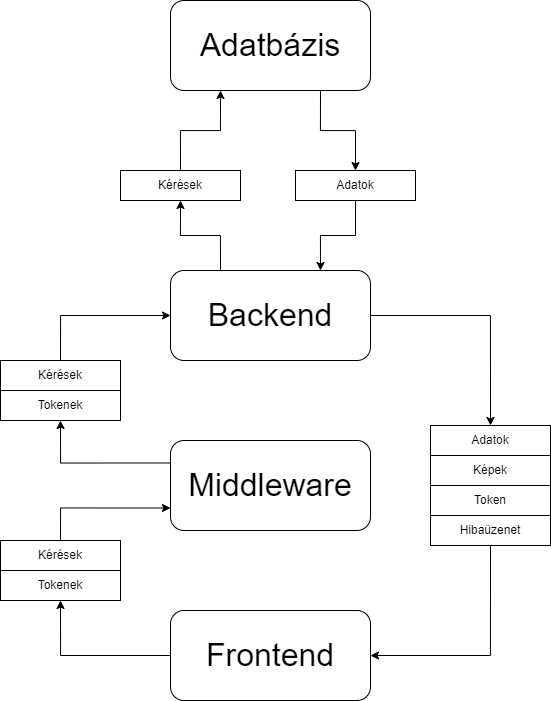
* Csaladom-Emleke/API/.env: tartalmazza az adatbázis adatait, az smtp szerver adatait (nodemailer) illetve a tokenek kódolásához szükséges kulcsot.
* Csaladom-Emleke/src/store.store.js: tartalmazza a weboldalon használt globális változókat.

Itt tároljuk az aktuális családfában szereplő személyeket

* Csaladom-Emleke/src/main.js: itt hozzuk létre **vue app**-ot és importáljuk be globálisan:
  + bootstrap
  + router
  + store
  + moment
* Csaladom-Emleke/index.html: ennek a html fájlnak a tartalmát a vue app szerkeszti és használja.
* Csaladom-Emleke/.gitignore: A GitHub-ra fel nem töltendő fájlokat soroljuk fel ebben a dokumentumban.
* Csaladom-Emleke/API/server.js: ebben a fájlban szerepelnek a szerver endpontjai valamint a tokenek ellenőrzése



**Adatbázis:**

**Kommunikáció:**

**Authorizáció (token):**

* A hitelesítés a **jsonwebtoken** node modul segítségével történik, amit a frontend oldal küld a backend részére
* Frontend oldalon ezt a localstorage-ban tároljuk (titkosított)
* 1 napos lejáratú a token
* egyszerre csak egy aktív token létezhet egy felhasználó számára.

**Kimenet:**

* Jó token: elérhető adatbázis kommunikáció
* Rossz token / Nincs token: kizárólag vendég funkciók érhetőek el.

**Tesztek:**

Megtalálhatóak a Dokumentáció mappában.