



FiveFile

VIZSGAREMEK DOKUMENTÁCIÓ

Csapat: **Pap András, Farkas Péter, Pardi Polett**
Gyakorlatvezető tanár neve: **Várkonyi Tibor**
Feladat megnevezése: **Vizsgaremek**

Tartalom

Szerző	1
Tartalom	2
Felhasználói dokumentáció	3
Feladat	3
Környezet	3
Használata	3
Alkalmazások	3
Bejelentkezés / Regisztráció kezdőlap	3
Menüpontok	5
Főoldal	5
Saját fájlok	7
Velem megosztva	7
Felhasználók	8
Beállítások	9
Kijelentkezés	9
Admin	10
Az oldal nem található	11
Fejlesztői dokumentáció	12
Megvalósítás	12
Forráskód	12
Backend	13
Adatbázis modell	13
Kontrollerek	15
Requestek	20
Resourcek	23
Modelek	25
Frontend	28
Útvonalak	28
Nézet fájlok	29
Komponensek	42
Android alkalmazás	48
A programozáshoz használt források jegyzéke	51

Felhasználói dokumentáció

Feladat

Egy online fájl tároló szoftvert készítünk, melynek célja, hogy a felhasználó egy helyen el tudja érni a számára fontos fájlokat. Regisztráció után a felhasználók tudnak fájlokat feltölteni, letölteni, törölni és megosztani egymás között.

Környezet

Az oldal bármilyen böngészőben futtatható. A fejlesztésnél többnyelvet is használtunk: Frontend HTML, CSS, Javascript, Vue JS, Php blade nyelvekkel lett elkészítve. A backend Php Laravelben lett megírva.

Használata

ALKALMAZÁSOK

Alkalmazások: GitHub, Docker, CMD/PowerShell/Terminal

Letöltjük a GitHub-ról a projektünket, majd elindítjuk a docker programot. Ezt követően a backend elindításához nem kell mást tenni, mint a Backend mappából elindítani az install kezdetű fájlok közül az általunk használt operációs rendszernek megfelelőt. Ha lefutott, átlépünk a Frontend mappába, és az általunk használt CLI programban megnyitjuk a Frontend mappát, és kiadjuk a következő: `npm install`, `npm run dev`. Ez után a CLI-ben megjelenő localhost linken meg tudjuk nyitni az alkalmazást.

Az android alkalmazás jelen állapotban csak az Android Studióból emulátorral futtatva tesztelhető.

BEJELENTKEZÉS / REGISZTRÁCIÓ KEZDŐLAP

Ha még nem regisztráltunk az oldalra, akkor egyértelműen ezt kell elsőként megtennünk. Meg kell adnunk egy felhasználónevet, emailcímet, jelszót majd egy jelszó ismétlést. Ha az email címet például @ nélkül töltjük ki, akkor kapni fogunk egy figyelmeztetést. Azt egyes beviteli mezőknek el kell érnie egy minimum szöveg hosszúság értéket, így a felhasználónévnek minimum 3 karakter hosszúnak, az email címnek minimum 10 karakter hosszúnak, a jelszónak pedig minimum 8 karakter hosszúnak kell lennie. A mezők jobb alsó sarkában az oldal számolja hogy hány karakter lett beütve. A regisztrálok gomb megnyomása után, átirányít minket a bejelentkezés felülethez.

A bejelentkezésnél írjuk be a már korábban regisztrált felhasználónevet, jelszót. Ha elfelejtettük a jelszót, tudunk kérni az Elfelejtett jelszó gombra kattintva egy új jelszót. Be kell írunk az email címünket, majd kapunk egy linket, amin keresztül tudunk új jelszót rögzíteni. Hibás adatok esetén az oldal erre is figyelmeztet.


5F

Bejelentkezés

Regisztráció

6/20

 Kérjük, írjon egy „@” karaktert az e-mail címbe. A(z) „pardipolettgmail.com” címből hiányzik a „@” jel.

8/20

8/20

Regisztrálok

Elfelejtett jelszó

Kérjük, add meg az e-mail címed, amellyel oldalunkra regisztráltál. A megadott címre küldünk egy linket, amelyre kattintva jelszavad módosíthatod!

Küldés

Hibás felhasználónév vagy jelszó!

MENÜPONTOK

Főoldal

<input type="checkbox"/>	Típus	Dokumentum neve	Méret	Legutóbbi módosítás	Tagok	#
<input type="checkbox"/>	📁	vue projekt	160 B	2022-04-29 11:48:37	Nincs megosztva senkivel	⋮
<input type="checkbox"/>	📄	fághgfhg	1.86 MB	2022-04-29 11:20:53	Nincs megosztva senkivel	⋮
<input type="checkbox"/>	📄	ehp.1408166.a001.acc4o	1.86 MB	2022-04-29 11:20:37	Nincs megosztva senkivel	⋮

Letöltés Megosztás Törölés

Hozz létre csoportot a megosztáshoz!

Típus	Méret
Zenék	0 B
Videók	2.16 MB
Képek	985.98 KB
Dokumentumok	224 B
Egyéb fájlok	5.59 MB
Összes fájl	8.72 MB / 10 MB

Fájl feltöltése

Fájlok kiválasztása Nincs új kiválasztva

Feltöltés

Az oldal bal szélén egy menüsáv található, ahol az alábbi menüpontok közül lehet választani: Főoldal, Saját fájlok, Velem megosztva, Felhasználók, Beállítások, Kijelentkezés.



A felhasználó, az oldalon láthatja, a legutóbbi fájljait, a feltöltött fájljairól készült statisztikát, mennyi helye maradt még, valamint egy fájl feltöltés részt. A menü felső részében megjelenik a felhasználó profilképe, melyet a beállításokban lehet feltölteni. A profil kép alatt látható a bejelentkezve eltöltött idő. Ez fél órára van állítva, és ha tétlenek vagyunk az oldalon, az idő leteltével automatikusan kijelentkeztetésre kerülünk.

<input type="checkbox"/>	Típus	Dokumentum neve	Méret	Legutóbbi módosítás	Tagok	#
<input type="checkbox"/>	📄	eating-habits-questionnaire-coach-version	193.93 KB	2022-04-29 10:12:05	Nincs megosztva senkivel	⋮

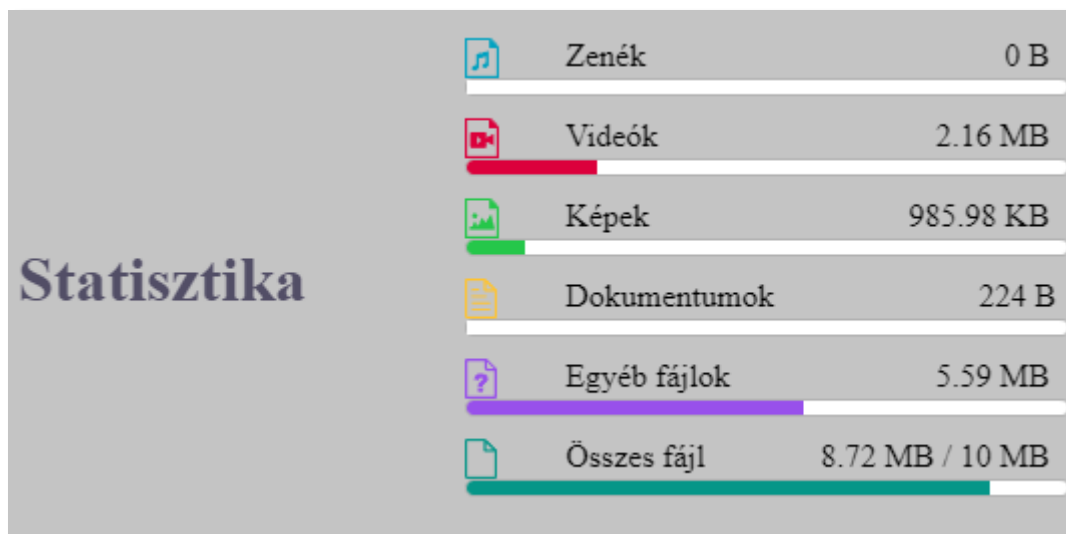
Letöltés Megosztás Törölés

Itt láthatjuk a 3 legfrissebben módosított fájlt, amit könnyedén letudunk tölteni, megosztani, valamint törölni az adott gombbal. Valamint megjelenik a dokumentum típushoz társított kép, a dokumentum neve, a mérete, a legutóbbi módosítás ideje, illetve, ha megosztottuk valakivel a megosztásban lévő felhasználók profilképei. A táblázat bal oldalán elhelyezkedő jelölő négyzetekkel egyszerre több fájlt is ki tudunk jelölni, melyekkel a jobb alul látható gombokkal tudunk műveleteket végezni. A táblázat fejlécében látható jelölőnégyzet segítségével az összes fájlt ki- és be tudjuk jelölni. Ha több fájlal végzünk műveletet, az oldal a gomb nyomása után a felhasználó megerősítését kéri. A táblázat sorainak jobb szélén látható egy kis menü ikon, melyre kattintva kinyílik az adott sor menüje, és a Letöltés-Megosztás-Törölés műveleteket el tudjuk végezni az adott

fájlon. A törlés opció kiválasztásakor az oldal megerősítést fog kérni.

<input type="checkbox"/>	Tipus	Dokumentum neve	Méret	Legutóbbi módosítás	Tagok	#
<input type="checkbox"/>		vue projekt	160 B	2022-04-29 11:48:37	Nincs megosztva senkivel	
<div><div>Letöltés Letöltés</div><div>Megosztás Hozz létre csoportot a megosztáshoz!</div><div>Törlés Törlés</div></div>						

A táblázat alatt bal oldalt láthatjuk a statisztika részt, ami megmutatja mennyi tárhely használtunk fel, valamint melyik dokumentum típus mennyi helyet foglal el.



A statisztika jobb oldalán pedig a fájlok feltöltése menü helyezkedik el. A beviteli mező legfeljebb 5 fájl feltöltését engedélyezi, 2 MB összesített méretig. Valamint, ha a fájlok mérete meghaladja a maradék rendelkezésre álló hely nagyságát, hibaüzenetet küld vissza.

Fájl feltöltése









Fájlok kiválasztása

Nincs fájl kiválasztva

Feltöltés

Saját fájlok

Saját fájlok

<input type="checkbox"/>	Típus	Dokumentum neve	Méret	Legutóbbi módosítás	Tagok	#
<input type="checkbox"/>		vue projekt	160 B	2022-04-29 11:48:37	Nincs megosztva senkivel	⋮
<input type="checkbox"/>		ehp.1408166.s001.acc4o	1.86 MB	2022-04-29 11:20:37	Nincs megosztva senkivel	⋮
<input type="checkbox"/>		fsdg	516.37 KB	2022-04-29 11:19:29	Nincs megosztva senkivel	⋮
<input type="checkbox"/>		ac	273.11 KB	2022-04-29 11:15:00	Nincs megosztva senkivel	⋮
<input type="checkbox"/>		artisan	64 B	2022-04-29 11:15:00	Nincs megosztva senkivel	⋮
<input type="checkbox"/>		AutoSave_1286f6_7e542800_43b7d9c	62.73 KB	2022-04-29 11:15:00	Nincs megosztva senkivel	⋮
<input type="checkbox"/>		42342	516.37 KB	2022-04-29 11:14:31	Nincs megosztva senkivel	⋮
<input type="checkbox"/>		2021124175319	203.49 KB	2022-04-29 11:14:31	Nincs megosztva senkivel	⋮

LetöltésMegosztásTörölés





Hozz létre csoportot a megosztáshoz!

Az oldalon a felhasználó saját fájljait listázzuk ki, ahol letudja tölteni, megosztani, valamint törölni őket.

Ugyan azok a funkciók érhetőek el, mint a főoldalon a legutóbbi fájlok táblázatában.





Velem megosztva

testgroup

Típus	Dokumentum neve	Méret	Tulajdonos	Letöltés
	ac	273.11 KB	papandras	<button>Letöltés</button>
	fsdg	516.37 KB	papandras	<button>Letöltés</button>
	ehp.1408166.s001.acc4o	1.86 MB	papandras	<button>Letöltés</button>
	vue projekt	160 B	papandras	<button>Letöltés</button>

[Kilépés a csoportból]

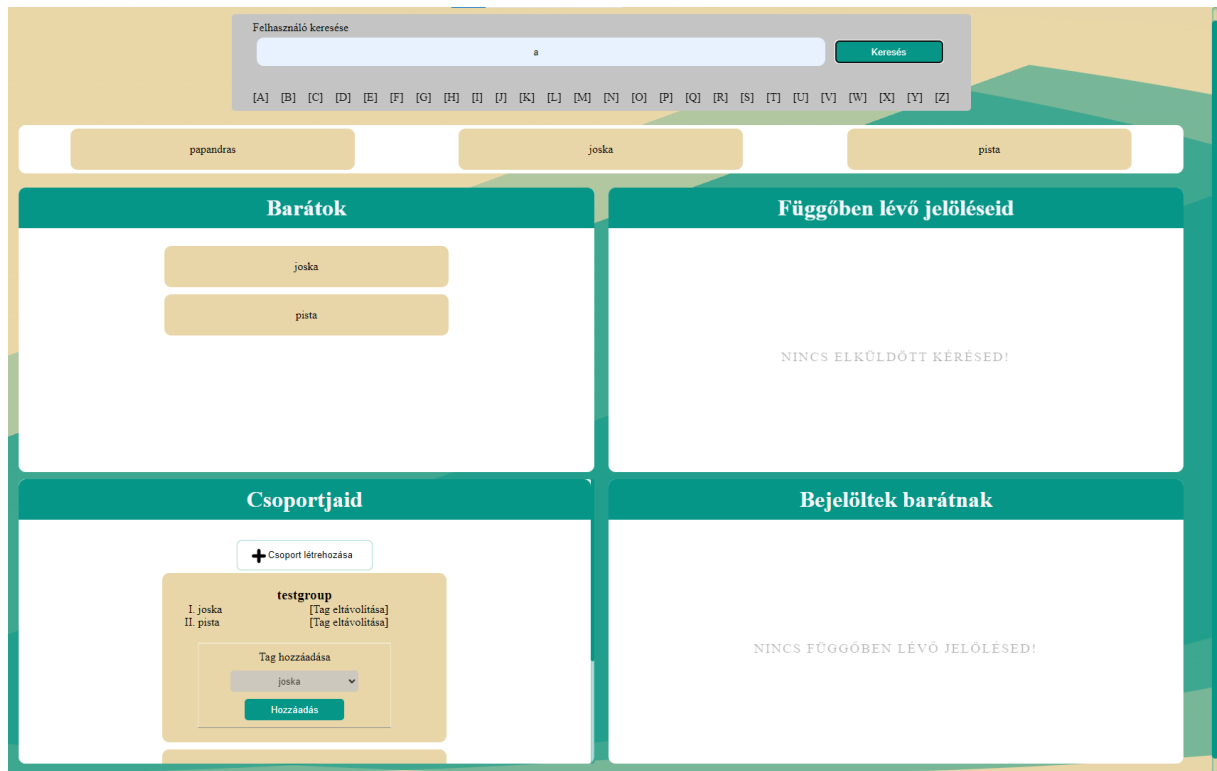
teszcsoport

Típus	Dokumentum neve	Méret	Tulajdonos	Letöltés
	42342	516.37 KB	papandras	<button>Letöltés</button>
	2021124175319	203.49 KB	papandras	<button>Letöltés</button>
	artisan	64 B	papandras	<button>Letöltés</button>
	AutoSave_1286f6_7e542800_43b7d9c	62.73 KB	papandras	<button>Letöltés</button>

[Kilépés a csoportból]

Ha tagja vagy már egy csoportnak akkor itt láthatod a veled megosztott fájlokat. A táblázat jelzi az adott fájl típusának az ikonját, a nevét, méretét, hogy kié a fájl, és az adott fájlt le is lehet tölteni. Esetlegesen lehetőség vagy a csoport elhagyására a [Kilépés a csoportból] gomb használatával.

Felhasználók



Az oldal tetején találunk egy keresés mezőt, itt rákereshetünk másik felhasználókra, majd be tudjuk őket jelölni. A kereső kidobja a saját profilunkat is, de természetesen magunkat nem tudjuk barátunk jelölni. A jelölés ikonja akkor jelenik meg, ha a név fölé visszük a kurzort. Tovább 4 kis dobozt látunk az oldalon: Barátaid, függőben lévő jelöléseid, csoportjaid, bejelöltek barátának. A csoportjaid dobozon belül tudsz létrehozni új csoportot, a barátokból új tagokat hozzáadni, és a jelenlegi tagokat el lehet távolítani. Végül a csoportokat ki is lehet törölni. Ennek ikonja a csoport doboz jobb alsó sarkában fog megjelenni. A barátokat el lehet távolítani (ikonja szintén akkor jelenik meg ha a név fölé visszük a kurzort). A Függőben lévő jelöléseid dobozban azok a felhasználók jelennek meg, akiket bejelöltünk barátunk, de még nem igazolták vissza a jelölést. Ezeket a jelöléseket vissza is lehet vonni a megfelelő ikonra kattintva. A Bejelöltek barátának az előbbinek az ellentéte. Itt azok a felhasználók láthatókat, akik minket jelöltek be, de még nem igazoltuk vissza. Itt lehetőség nyílik a jelölés elutasítására is.

Beállítások

Adatok

Biztonság

Új kép feltöltése

Fájl kiválasztása Nincs fájl kiválasztva ☐ Jelenlegi törlése

E-mail cím

Jelszó megerősítése Módosítások mentése

Adatok

Jelszó módosítása

Fiók törlése

Ez a művelet nem visszavonható!

Jelszó megerősítése Módosítások mentése

Az Adatok gombra kattintva, megjelenik az adatok beállítások menüpontja. Itt a felhasználó tudja a profilképét és az emailcímét is módosítani. Ezt a jelenlegi jelszó beírásával tudja elmenteni. Ha van beállított profilkép akkor a jelenlegi törlése jelölő négyzet bepipálásával lehet eltávolítani.

A biztonság gombra kattintva a jelszavát tudja módosítani, valamint a fiókját tudja véglegesen törölni. Bár két menüpont van, de a mentés gomb mind a kettőre egyszerre érvényes.

Kijelentkezés



A gombra kattintva a felhasználó kijelentkezik a weboldalról. Megjelenik egy sikeres kijelentkezés felirat, amely alatt a számláló 5-től másodpercenként visszaszámol, 1 után pedig átirányít a bejelentkezés oldalra.

Admin

Fájlok kezelése
Felhasználók kezelése

Fájl neve
Felhasználó neve

Keresés

Itt tudsz szűrni fájlt és/vagy felhasználó névre. Ha nem adsz meg semmi akkor mindent kilistáz!

Név	Kiterjesztés	Méret	Tulajdonos	Feltöltve	Legutóbb módosítva	Letöltés	Törölés
42342	mp4	516.37 KB	papandras	2022-04-29 11:14:31	2022-04-29 14:23:33	Letöltés	Törölés
2021124175319	jpg	203.49 KB	papandras	2022-04-29 11:14:31	2022-04-29 14:23:37	Letöltés	Törölés
ac	psd	273.11 KB	papandras	2022-04-29 11:15:00	2022-04-29 14:23:14	Letöltés	Törölés
artisan	txt	64 B	papandras	2022-04-29 11:15:00	2022-04-29 14:23:33	Letöltés	Törölés
AutoSave_1286f6_7e542800_43b7d9c	png	62.73 KB	papandras	2022-04-29 11:15:00	2022-04-29 14:23:33	Letöltés	Törölés
fsdg	mp4	516.37 KB	papandras	2022-04-29 11:19:29	2022-04-29 14:23:11	Letöltés	Törölés
chp.1408166.s001.acc4o	pdf	1.86 MB	papandras	2022-04-29 11:20:37	2022-04-29 14:23:11	Letöltés	Törölés
vue projekt	txt	160 B	papandras	2022-04-29 11:48:37	2022-04-29 14:23:10	Letöltés	Törölés
Képernyőfelvétel (934)	png	182.66 KB	joska	2022-04-29 14:27:00	2022-04-29 14:27:00	Letöltés	Törölés

Fájlok kezelése
Felhasználók kezelése

Add meg a keresett felhasználó nevét:

Keresés

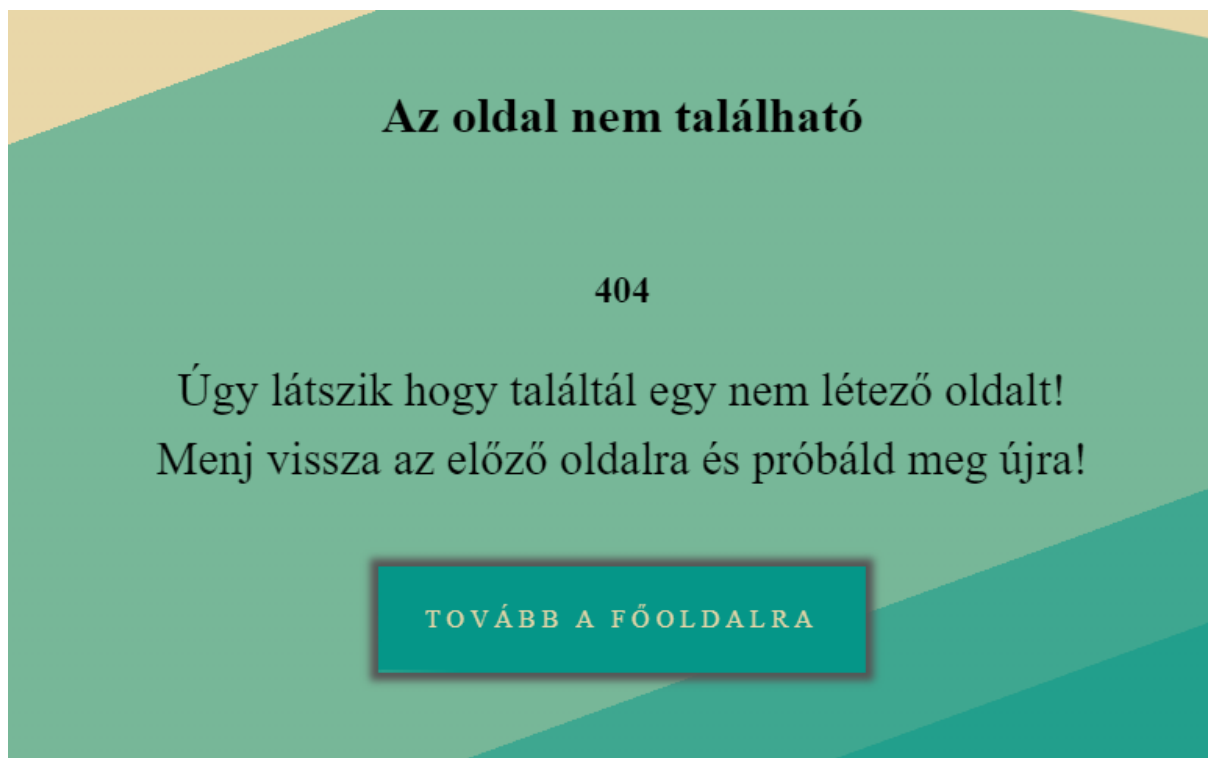
[Összes megjelenítése]

Név	Email	Tárhely foglaltsága	Regisztrált	Szerepkör módosítása Jelenlegi: admin	Törölés
polli	polli@example.com	0 B	2022-04-26 19:34:34	Legyen user	Törölés
peti	peti@example.com	0 B	2022-04-26 19:34:34	Jelenlegi: admin Legyen user	Törölés
joska	joska@gmail.com	182.66 KB	2022-04-28 21:45:33	Jelenlegi: user Legyen admin	Törölés
pista	pista@gmail.com	0 B	2022-04-29 14:17:02	Jelenlegi: user Legyen admin	Törölés

A menüpont nevéből adódóan ez a menüpont csak admin jogosultsággal rendelkező felhasználók által tekinthető meg. Itt lehet a felhasználókat és a fájlokat (az összes felhasználó fájljait) kezelni. A felhasználóknak lehet jogosultságot állítani (admin/user), és ki lehet őket törölni. Tartozik a menüponthoz egy felhasználó keresés is. Akár egy betűre is lehet keresni, és kidobja az beleillő nevű felhasználókat.

A fájloknál egy szinttel komplexebb a keresés funkció. Lehet a fájl nevére keresni, és a felhasználók fájljaira keresni, és ezek kombinációjára. Ha nem töltjük ki egyiket sem a keresés gomb az összes adatot megjeleníti. Ha csak fájl névre keresünk akkor az összes beleillő nevű fájl között keres. Ha csak felhasználó névre keresünk akkor az adott felhasználó összes fájlját megjeleníti. Ha mindkét feltétellel keresünk, akkor az adott felhasználó adott nevű fájljait jeleníti meg.

Az oldal nem található



Az oldal gondoskodik azokról a felhasználókról is, akik olyan linkre keverednek, amely nem található. Ilyenkor megjelenik a fent látható képernyő, és a gombra kattintva el lehet jutni a főoldalra.

Fejlesztői dokumentáció

Megvalósítás

BACKEND

A projekt backendjéül egy Laravel alkalmazás szolgál.

FRONTEND

A projekt frontendje VUE JS keretrendszerrel lett megvalósítva.

ANDROID

A mobil alkalmazás a Google által fejlesztett Flutterrel lett elkészítve

Forráskód

A projekthez tartozó teljes fejlesztői anyag megtalálható az alábbi linken:

- <https://github.com/papandras/NJIT-2021-2022-Vizsgaremek>

A projektet letöltve a további almappokban érhetőek el a projekt részegységei.

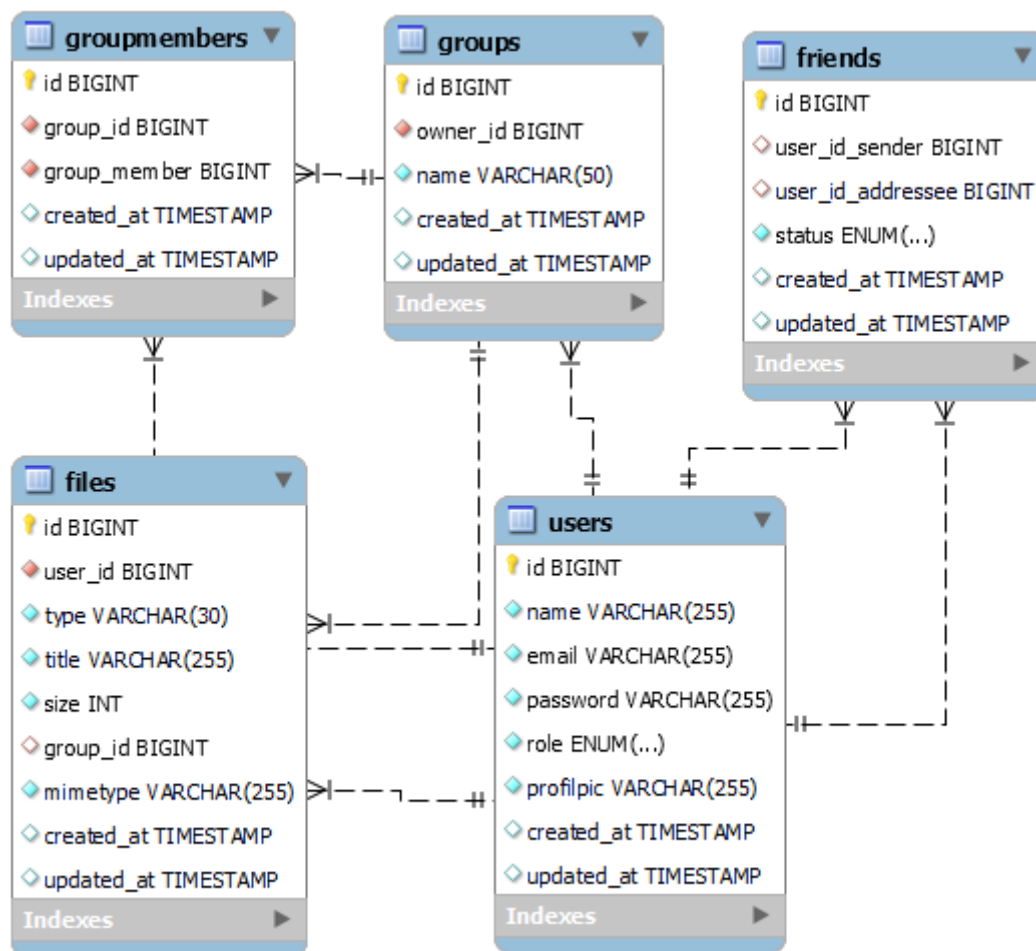
Futtatáshoz szükséges információk:

- Backend: install.bat / install.sh / install.ps1 fájl futtatára.
 - Ezt követően az alkalmazás elérhető a localhost:8881-en, az adatbázis pedig a localhost:8882-n
- Frontend:
 - a következő parancsok kiadása szükséges:
 - npm install
 - npm run dev / npm run build
 - Ezt követően az alkalmazás elérhető a CLI által jelzett címen (localhost:3000)
- Android: android studio ide-ből emulátorral futtatható
- A parancsok futtatásakor az adott projekt főmappája tekintendő root könyvtárnak!

Backend

Adatbázis modell

A fájlok láthatóságának esetében lényeges feladat volt az adatbázis megtervezése. A modell elkészítését követően, a modell alapján írt migrációk segítségével le tudtam kérdezni a megfelelő rekordokat. Ezt követően a felhasználó azokat a fájlokat kapta vissza, amelyeket kizárólag Ő töltött fel, illetve egy másik felhasználó által meg lettek osztva vele. A tervezés alapja a felhasználó adatainak tárolása, amik az *users* táblában szerepelnek. A *users* táblához 3 tábla kapcsolódik közvetlenül, és 1 közvetetten. Minden felhasználóhoz tartoznak a saját fájljai, a barátai, és az általa létrehozott csoportjai. Ezek az adatok rendre a *files*, *friends* és *groups* táblákban találhatóak. A csoportokhoz természetesen lehet tagokat is hozzárendelni. Ezek a tagok a *groupmembers* táblában vannak eltárolva. A csoport rendszer közvetlen összefüggésben áll a barát rendszerrel, mivel a csoport tagjai csak olyan felhasználók lehetnek, akik az adott felhasználó barát listáján szerepelnek. A csoport rendszer pedig a fájlok megosztását létesíti meg. Fájlokat nem közvetlenül lehet a többi felhasználóval megosztani, hanem egy csoporton keresztül.



Ábra: Adatbázis EER modell 1

Az adatbázishoz tartoznak a fent említetteken kívül további táblák is. A *migrations* táblára a Laravel keretrendszernek van szüksége. Ebben vannak a lefuttatott migrációk tárolva. A *personal_access_tokens* táblában a felhasználók bejelentkeztetéséért felelős *token*-ek kapnak helyet. A felhasználóknak továbbá lehetőségük van a jelszó visszaállítására, amennyiben nem tudnak bejelentkezni az oldalra. Ebben az esetben az email címükre küldött linken keresztül képesek új jelszót beállítani.

personal_access_tokens	migrations	password_resets
<ul style="list-style-type: none">id BIGINTtokenable_type VARCHAR(255)tokenable_id BIGINTname VARCHAR(255)token VARCHAR(64)abilities TEXTlast_used_at TIMESTAMPcreated_at TIMESTAMPupdated_at TIMESTAMP <div>Indexes ▶</div>	<ul style="list-style-type: none">id INTmigration VARCHAR(255)batch INT <div>Indexes ▶</div>	<ul style="list-style-type: none">email VARCHAR(255)token VARCHAR(255)created_at TIMESTAMP <div>Indexes ▶</div>

Ábra: Adatbázis EER modell 2

Controller-ek

Az alábbiakban ismertetésre kerülnek az alkalmazás által használt kontrollerek, illetve az ezek elérésére használt útvonalak.

Az alkalmazás az alábbi táblázatokban található útvonalak elérését biztosítja. A Method oszlopban az útvonal metódusa, az URI oszlopban az útvonal, Action oszlopban az útvonalhoz tartozó @ jellel ellátott controller metódus, a Name oszlopban az útvonal neve, a Middleware oszlopban pedig az útvonalhoz tartozó szűrő található. Minden metódus a fejlécben jelzett controller osztályhoz tartozik.

A táblázatok alatt található listában az adott táblázatban szereplő metódusok rendeltetésére lehet kapható magyarázat.

App\Http\Controllers\AuthController				
Method	URI	Action	Name	Middleware
POST	api/register	@register	register	-
POST	api/login	@login	login	-
POST	api/forgot-password	@forgotpassword	password.request	-
GET	api/user	@user	user	auth:sanctum
GET	api/getNewCookie	@getNewCookie	getNewCookie	auth:sanctum
POST	api/logout	@logout	logout	auth:sanctum
GET	resetpassword/{token}	@resetpassword	password.reset	guest
POST	update-password	@updatepassword	password.update	guest

- @register: Ez a metódus a felhasználói fiók elkészítésére szolgál.
- @login: Ez a metódus a felhasználónév – jelszó párossal történő bejelentkezést biztosítja.
- @logout: Kijelentkezteti a felhasználót.
- @forgotpassword: Sikeres email cím megadása esetén a megadott email címre küld üzenetet, melyen keresztül elérhető számára a jelszó visszaállítására szolgáló felület.
- @resetpassword: A kiküldött emailen talál található linket biztosítja. Erre kattintva érhető el a jelszó visszaállítására szolgáló felület.
- @updatepassword: A megadott új jelszót tárolja el az adatbázisban.
- @user: A bejelentkezett felhasználó adatait adja vissza.
- @getNewCookie: A bejelentkezett felhasználó számára új sütit biztosít. A felhasználói felület újra töltésekor lép életbe. A süti élettartama 30 perc, és ennek meghosszabbítását szolgálja.

App\Http\Controllers\UserController				
Method	URI	Action	Name	Middleware
GET	api/users	@getAllUsers	getAllUsers	auth:sanctum
GET	api/users/{username}	@getUser	getUser	auth:sanctum
GET	api/users/letter/{startletter}	@getUserByStartLetter	getUserByStartLetter	auth:sanctum
GET	api/user/{id}/groups	@getgroups	user.groups	auth:sanctum
GET	api/user/groups	@getownedgroups	user.owngroups	auth:sanctum
GET	api/group/{id}/files	@getFilesInGroup	group.files	auth:sanctum

- @getAllUsers: Visszatér az összes regisztrált felhasználó adataival a UserResource osztályban meghatározott szabályok szerint.
- @getUser: Visszatér a *username* paraméterben megkapott felhasználó adataival a UserResource osztályban meghatározott szabályok szerint.
- @getUserByStartLetter: Visszatér az összes olyan felhasználóval, akiknek felhasználóneve a *startletter* paraméterben megkapott betűvel kezdődik.
- @getgroups: Visszatér az összes olyan csoport adataival, amelynek tagjaihoz hozzá van rendelve az adott felhasználó.
- @getownedgroups: Visszatér a felhasználó összes saját csoportjával.
- @getFilesInGroup: Visszatér a paraméterben megkapott csoportban található összes fájlal a GroupFileResource osztályban meghatározott szabályok szerint.

App\Http\Controllers\FriendController				
Method	URI	Action	Name	Middleware
GET	api/user/friends	@getFriends	friends	auth:sanctum
POST	api/user/friends/add	@addFriend	addfriend	auth:sanctum
POST	api/user/friends/remove	@removeFriend	removeFriend	auth:sanctum
GET	api/user/friends/friendrequests	@friendrequests	friendrequests	auth:sanctum
POST	api/user/friends/accept	@acceptrequest	acceptrequest	auth:sanctum
GET	api/user/friends/sentrequests	@sentrequests	sentrequests	auth:sanctum
POST	api/user/friends/cancelrequest	@cancelrequest	cancelrequest	auth:sanctum
POST	api/user/friends/rejectrequest	@rejectrequest	rejectrequest	auth:sanctum

- @getFriends: Visszatér a felhasználó összes megerősített barátjával.
- @addFriend: Barát jelölést küld az adott felhasználónak. Requestben kötelező elküldeni a bejelölendő barát nevét!
- @removeFriend: Megszünteti a barát kapcsolatot az adott felhasználóval. Requestben kötelező elküldeni a töröltető barát azonosítóját!

- @friendrequests: Visszatér az adott felhasználó összes olyan bejövő barát kérésével, amely még nem lett elfogadva.
- @acceptrequest: Az adott barát kérés megerősítésére szolgál.
- @sentrequests: Visszatér az adott felhasználó összes olyan elküldött barát kérésével, amely még nem lett elfogadva.
- @cancelrequest: Adott elküldött és még nem visszaigazolt barát kérés visszavonása. Requestben kötelező elküldeni a visszavonandó kérés azonosítóját!
- @rejectrequest: Bejövő barát kérés elutasítása. Requestben kötelező elküldeni az elutasítandó kérés azonosítóját!

App\Http\Controllers\GroupController				
Method	URI	Action	Name	Middleware
POST	api/group/create	@create	group.create	auth:sanctum
POST	api/group/addmember	@addmember	group.addmember	auth:sanctum
GET	api/group/{id}/members	@getmembers	group.members	auth:sanctum
DELETE	api/group/{id}/delete	@deletegroup	group.delete	auth:sanctum
DELETE	api/group/{group}/member/{user}/delete	@deletememberfromgroup	deletememberfromgroup	auth:sanctum
DELETE	api/group/{id}/leave	@leavegroup	group.leave	auth:sanctum

- @create: Csoport létrehozása. A requestben kötelező szerepelnie a *name* paraméternek!
- @addmember: Barát hozzáadása adott csoporthoz. A requestben kötelező elküldeni a barát és a csoport azonosítóját!
- @getmembers: Az *id* paraméterben átadott csoport tagjait adja vissza.
- @deletegroup: Az *id* paraméterben átadott csoportot törli az adatbázisból.
- @deletememberfromgroup: A *group* paraméterben megadott csoportból törli a *user* paraméterben megadott felhasználót.
- @leavegroup: Az *id* paraméterben megadott csoportból törli a requestet küldő felhasználót.

App\Http\Controllers\FileController				
Method	URI	Action	Name	Middleware
POST	api/file/upload	@upload	file.upload	auth:sanctum
GET	api/file/get/{limit?}	@getFiles	file.get	auth:sanctum
GET	api/file/download/{filename}	@download	file.download	auth:sanctum
DELETE	api/file/delete/{filename}	@delete	file.delete	auth:sanctum
PUT	api/file/{id}	@sharewithgroup	file.share	auth:sanctum
PUT	api/file/{id}/unshare	@unshare	file.unshare	auth:sanctum
GET	api/file/stat	@getstat	file.stat	auth:sanctum

- @upload: A requestben érkező fájlokat eltárolja a projekt *storage/app/storage* mappájában a következő néven: *felhasználónév-fájlnev.kiterjesztés*. Ugyan ezen a néven eltárolja az adatbázis *files* táblájába az Adatbázis modellben látható adatokkal együtt.
- @getFiles: Visszatér a felhasználó fájljaival. A *limit* paraméter opcionális. Ha nincs megadva akkor az összes fájlt visszaadja, ha meg van adva akkor annyi darabszámú lesz csak visszaadva.
- @download: A *filename* paraméterben megkapott fájl letöltését teszi elérhetővé.
- @delete: A *filename* paraméterben megkapott fájl törlését teszi lehetővé.
- @sharewithgroup: Az *id* paraméterben átadott fájlt osztja meg a requestben megkapott *group_id* id-jú csoporttal.
- @unshare: Az *id* paraméterű fájl megosztását vonja vissza.
- @getstat: Az adott felhasználó tárolt fájljairól készít statisztikát fájl típusonként és összességében.

App\Http\Controllers\AdminController				
Method	URI	Action	Name	Middleware
GET	api/admin/users/{user?}	@getUsers	admin.finduser	auth:sanctum
PUT	api/admin/users/setrole/{user}/{role}	@setrole	admin.setrole	auth:sanctum
DELETE	api/admin/users/delete/{user}	@deleteuser	admin.deleteuser	auth:sanctum
POST	api/admin/files	@getFiles	admin.getfiles	auth:sanctum

- @getUsers: Adott felhasználó keresésére szolgál. A *user* paraméter opcionális. Ha nincs kitöltve az összes felhasználót visszaadja, ha ki van töltve akkor csak a paraméterbe illő nevű felhasználókat adja vissza.
- @setrole: A *user* paraméterű felhasználó szerepkörét állítja a *role* paraméterben megkapott szerepkörre. Két fajta szerepkör van: admin, user.
- @deleteuser: A *user* paraméterben megkapott felhasználót törli.
- @getFiles: A neki küldött requestnek 2 paraméter adható át: *user* és *file*. Egyik paraméter sem kötelező. Ha egyik sincs kitöltve akkor minden fájlt visszaad. Ha csak a *user* van kitöltve

akkor az adott felhasználó összes fájlját visszaadja. Ha a *file* is ki van töltve akkor az adott felhasználó olyan fájljait adja vissza, amelyek illenek a *file* paraméterű kulcsszóra. (szabad szavas keresés). Ha csak a *file* van kitöltve, akkor felhasználótól függetlenül adja vissza az illeszkedő nevű fájlokat.

App\Http\Controllers\SettingsController				
Method	URI	Action	Name	Middleware
POST	api/user/settings	@settings	user.settings	auth:sanctum
DELETE	api/user/delete	@removeaccount	user.delete	auth:sanctum

- @settings: A következő adatok megváltoztatását teszi lehetővé: (a hozzá szükséges request paraméter)
 - Email cím (*email*)
 - Jelszó (*password*)
 - Profil kép (*profilpic / removepic*)
 - A requestben minden esetben kötelező elküldeni a *confirm* paramétert, amely a jelenlegi jelszó megerősítésére szolgál.
- @removeaccount: A kérést küldő felhasználó profilját törli, majd kijelentkezteti az oldalról.

Request-ek

A Laravel Request osztálya segítségével hozzá tudunk férni a bejövő kérések adataihoz, többek között a sütikhez, és egyéb fájlokhoz. Továbbá lehetőségünk van egyéni Request osztályok létrehozására, melyekben le tudjuk kezelni, hogy az aktuális felhasználó végre tudja-e hajtani az adott kérést, és ha igen, a bejövő adatokra tudunk szabályokat kiszabni. Továbbá lehetőségünk van az alapértelmezett angol üzeneteket felülrírni. Ezt az osztályok messages metódusában tudjuk majd megtenni, ahogy majd látható a RegisterRequest és a PasswordResetRequest osztályokban.

A FiveFile alkalmazásban 3 egyéni request osztály található.

1) LoginRequest

Alkalmazott szabályok:

- name: kötelező kitölteni, minimum 3 és maximum 20 karakter hosszú.
- password: kötelező kitölteni, minimum 8 és maximum 20 karakter hosszú.

```
class LoginRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize( )
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules( )
    {
        return [
            'name' => 'required|min:3|max:20',
            'password' => 'required|min: 8|max:20',
        ];
    }
}
```

Ábra: LoginRequest

2) RegisterRequest

Alkalmazott szabályok:

- name: kötelező kitölteni, minimum 3 és maximum 20 karakter hosszú.
- email: kötelező kitölteni, nem szerepelhet már a *users* tábla *email* oszlopában. Hossza 10 és 50 karakter között lehet.
- password: kötelező kitölteni, minimum 8 és maximum 20 karakter hosszú.

```
class RegisterRequest extends FormRequest
{
  /**
   * Determine if the user is authorized to make this request.
   *
   * @return bool
   */
  public function authorize()
  {
    return true;
  }

  /**
   * Get the validation rules that apply to the request.
   *
   * @return array
   */
  public function rules()
  {
    return [
      'name' => 'required|unique:users,name|min:3|max:20',
      'email' => 'required|unique:users,email|min:10|max:50',
      'password' => 'required|min:8|max:20',
    ];
  }

  public function messages()
  {
    return [
      'name.required' => 'A felhasználónév megadása kötelező!',
      'name.unique' => 'Ezzel a felhasználónévvel már regisztráltak!',
      'name.min' => 'A felhasználónév minimum 3 karakter hosszú!',
      'name.max' => 'A felhasználónév maximum 20 karakter hosszú!',
      'email.required' => 'Az email megadása kötelező!',
      'email.unique' => 'Ezzel az email címmel már regisztráltak!',
      'email.min' => 'Az email minimum 3 karakter hosszú!',
      'email.max' => 'Az email maximum 20 karakter hosszú!',
      'password.required' => 'A jelszó megadása kötelező!',
      'password.min' => 'A jelszó minimum 8 karakter hosszú!',
      'password.max' => 'A jelszó maximum 20 karakter hosszú!',
    ];
  }
}
```

Ábra: RegisterRequest

3) PasswordResetRequest

Alkalmazott szabályok:

- name: kötelező kitölteni, minimum 3 és maximum 20 karakter hosszú.
- password: kötelező kitölteni, minimum 8 és maximum 20 karakter hosszú, meg van erősítve.
- password_confirmation: kötelező kitölteni

```
class PasswordResetRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'email' => 'required|email|exists:users,email',
            'password' => 'required|string|min:8|confirmed',
            'password_confirmation' => 'required'
        ];
    }

    public function messages()
    {
        return [
            'password.confirmed' => 'A jelszavak nem egyeznek!',
            'email.exists' => 'Nincs ilyen regisztrált email cím!'
        ];
    }
}
```

Ábra: PasswordResetRequest

Resources

A Laravel Resource osztályokkal könnyedén tudjuk az adatbázis lekérdezések eredményeit átalakítani JSON formátumba, továbbá meg tudjuk adni, hogy milyen formában kapjuk vissza az adatokat. Ez akkor előnyös, ha különböző felhasználói csoportoknak nem akarunk minden egyes adatot megjeleníteni.

Resource osztályt a php artisan segítségével tudunk generálni a következő parancs kiadásával:

```
php artisan make:resource NameResource
```

A generált osztályok az app/Http/Resources mappába fognak létrejönni.

A FiveFile alkalmazásban 6 resource osztály található.

1) AdminFileResource

```
return [
    "name" => $filename,
    "type" => $this->type,
    "size" => $filesize,
    "owner" => $ownername->name,
    "uploaded" => Carbon::parse($this->created_at)->timezone('Europe/Budapest')->toDateTimeString(),
    "edited" => Carbon::parse($this->updated_at)->timezone('Europe/Budapest')->toDateTimeString(),
];
```

2) AdminUserResource

```
return [
    "id" => $this->id,
    "name" => $this->name,
    "email" => $this->email,
    "filesize" => $filesize,
    "registered" => Carbon::parse($this->created_at)->timezone('Europe/Budapest')->toDateTimeString(),
    "role" => $this->role
];
```

3) FileResource

```
return [
    "id" => $this->id,
    "owner_id" => $this->user_id,
    "type" => $this->type,
    "name" => $filename,
    "size" => $filesize,
    "shared_group_id" => $this->group_id,
    "created" => Carbon::parse($this->created_at)->timezone('Europe/Budapest')->toDateTimeString(),
    "updated" => Carbon::parse($this->updated_at)->timezone('Europe/Budapest')->toDateTimeString(),
];
```

4) FriendResource



```
return [  
    "id" => $this->id,  
    "nev" => $this->name,  
    "email" => $this->email,  
    "regisztralt" => $this->created_at,  
];
```

5) GroupFileResource



```
return [  
    "id" => $this->id,  
    "owner" => $owneruser->name,  
    "type" => $this->type,  
    "name" => $filename,  
    "size" => $filesize,  
    "shared_group_id" => $this->group_id,  
    "created" => Carbon::parse($this->created_at)->timezone('Europe/Budapest')->toDateTimeString(),  
    "updated" => Carbon::parse($this->updated_at)->timezone('Europe/Budapest')->toDateTimeString(),  
];
```

6) UserResource



```
return [  
    "id" => $this->id,  
    "nev" => $this->name,  
    "email" => $this->email,  
    "regisztralt" => $this->created_at,  
    "rang" => $this->role,  
    "kep" => $this->profilpic  
];
```


Modellek

A Laravel projektben a model osztályok az app/Models mappában helyezkednek el. Az eddigiekhez hasonlóan a modelleket is a php artisan-nal lehet generálni:

```
php artisan make:model NameModel
```

Az alkalmazásban 5 model található.

1) User

```
class User extends Authenticatable
{
    use HasFactory, Notifiable, HasApiTokens;

    protected $fillable = ['name', 'email', 'password', 'profilpic'];

    protected $hidden = [
        'password'
    ];

    public function friends()
    {
        return $this->belongsToMany(User::class, 'friends', 'user_id_sender', 'user_id_addressee');
    }
}
```

2) File

```
class File extends Model
{
    use HasFactory;

    protected $fillable = [
        "user_id", "type", "title", "size"
    ];
}
```

3) Friend



```
class Friend extends Model
{
    use HasFactory;

    protected $table = "friends";

    protected $fillable = ["user_id_sender"];
}
```

4) Group



```
class Group extends Model
{
    use HasFactory;

    protected $fillable = [
        "owner_id", "name"
    ];
}
```

5) GroupMember



```
class GroupMember extends Model
{
  use HasFactory;

  protected $table = "groupmembers";

  protected $fillable = ['group_id', 'group_member'];
}
```

Frontend

A FiveFile alkalmazás frontend része Vue JS keretrendszerrel lett megvalósítva. Maga az alkalmazás egy SPA (Single Page Application), ami annyit tesz, hogy minden adat az index.html-re renderelődik ki, és az router segítségével lehet az oldalakon mozogni. Az oldal forráskódja az src mappában található. Az src-n belül 5 almappába vannak csoportosítva a fájlok. Az assets mappában van az összes megjelenő ikon - a kategóriák ikonjai és fájl kiterjesztés ikonjai. Továbbá adott komponensek js és css fájljai az ezekre kijelölt további almappákban. Következő mappa a components. Ebben a mappában találhatóak az oldalon használt komponensek. Működésük a későbbiekben kerül ismertetésre. A router mappában található index.js felel az alkalmazásban található útvonalakért. Itt vannak meghatározva, hogy melyik útvonal melyik view fájlt töltsse be, és vizsgálja, hogy a bejelentkezett felhasználónak van-e jogosultsága megtekinteni azt. Amennyiben nincsen bejelentkezett felhasználó, akkor nem tudja megtekinteni az oldalakat, mivel nincsen vendég mód, a használat regisztrációhoz kötött. A router gondoskodik még arról is, hogy ha a felhasználó nem létező útvonalat próbál elérni, hibaüzenetet kap. A router mappát követően a store mappában lévő auth.js felel az autentikáció lebonyolításáért. Megvizsgálja, hogy az oldalon elérhető a backend által küldött süti. Amennyiben elérhető akkor a felhasználó be van jelentkezve, ha pedig nem, akkor a router átirányítja a bejelentkező oldalra. Ha pedig be van jelentkezve, akkor a bejelentkező oldalról irányít át az index oldalra. Maga a store a pinia használatával került megvalósításra. Ennek hivatkozása megtalálható a dokumentáció végén. A következő views mappában az oldal nézet fájljai találhatóak. Mind a 11 útvonalhoz 1-1 nézet tartozik. Az első nézet amivel a felhasználó találkozik, az az LoginView.vue fájlban található. Itt van a bejelentkező felület, illetve a regisztrációs oldalra is innen lehet tovább lépni. A továbbiakban részletezve lesz, hogy milyen útvonalak találhatóak az alkalmazásban, és a komponensek, amelyekből felépül.

Path: útvonal, Title: a menüpont neve, View: a betöltődő nézet fájl neve, RequiresAuth: szükséges-e hozzá bejelentkezés

Útvonalak

Path	Title	View	RequiresAuth
/	Bejelentkezés	LoginView.vue	false
/register	Regisztráció	RegisterView.vue	false
/index	Főoldal	IndexView.vue	true
/myfiles	Saját fájlok	MyFilesView.vue	true
/shared	Velem megosztott fájlok	SharedView.vue	true
/users	Felhasználók	UsersView.vue	true
/admin	Admin	AdminView.vue	true + admin role
/settings	Beállítások	SettingView.vue	true
/unauthorized	Nem jelenkeztél be	Unauthorized.vue	false
/logout	Kijelentkezés	LogoutPage.vue	true
/:pathMatch(.*)*	404	NotFoundPage.vue	true

Nézet fájlok

LoginView



```
<template>
  <div class="login">
    <LoginPage></LoginPage>
  </div>
</template>
```

Használt komponensek: Login (Login.vue)

RegisterView



```
<template>
  <div class="login">
    <RegisterPage></RegisterPage>
  </div>
</template>
```

Használt komponensek: Register (Register.vue)

IndexView



```
<template>
  <div id="indexpage">
    <IndexPage />
  </div>
</template>
```

Használt komponensek: IndexPage (Index.vue)

MyFilesView

```
<template>
  <div class="myfiles">
    <Menu id="menu" activepage="owned"></Menu>
    <div id="content">
      <FileTable :lastfilesobject="files" :refresh="getlastfiles" title="Saját fájlok" id="myfiles"
name="myfiles"/>
    </div>
  </div>
</template>
```

Használt komponensek

Menu (LeftSideMenu.vue)

- Ez a komponens alkotja magát az oldal menüjét.
- Az oldalak fő szerkezete CSS grid-del van megvalósítva. Egy sor 1:9 arányban van mindig felosztva. A bal oldali oszlopban a menüsor, a jobb oldaliban pedig az oldal további tartalma.

FileTable (LastFilesTable.vue)

- Ebben a táblázatban vannak megjelenítve a felhasználó fájljai.

Állapotok

data

files

- Kezdőértéke: null
- Ide fognak betöltődni a felhasználó fájljainak az adatai, majd át lesz adva a FileTable komponensnek, amely megjeleníti őket.

methods

getfiles

- Lekéri a bejelentkezett felhasználó fájljait a backend api/file/get útvonaláról, majd átadja őket a files állapotnak.

SharedView

```
<div v-if="groups != null">
  <div class="group" v-for="group in groups" :key="group.id">
    <h1>{{ group.name }}</h1>
    <table :id="group.id + '-table'">
      <tr class="thead">
        <th>Típus</th>
        <th>Dokumentum neve</th>
        <th>Méret</th>
        <th>Tulajdonos</th>
        <th>Letöltés</th>
      </tr>
      <TableRow :groupid="group.id" :nofilemessage="
        () => {
          nofilemessage = true;
        }
      " />
      <tr>
        <td colspan="4"></td>
        <td><p class="leavegroup" @click="leavegroup(group.id)">[Kilépés a csoportból]</p></td>
      </tr>
    </table>
    <div class="nofilemessage" :id="group.id + '-nofile'">
      A csoportban nem található fájl!
    </div>
  </div>
</div>
<div v-if="groups == null" class="nofilemessage">
  Nem vagy tagja egy csoportnak sem!
</div>
```

Ez az oldal jeleníti meg azokat a csoportokat, amelyekhez az adott felhasználó hozzá lett adva. Amennyiben egy ilyen csoport sincs, kiírja, hogy „Nem vagy tagja egy csoportnak sem!”. Ha vannak csoportok, akkor a v-for direktíva segítségével kilistázza őket, és a benne lévő fájlokat. Ha nincsen fájl a csoportban akkor is kap a felhasználó üzenetet. Továbbá lehetőség van kilépni az adott csoportból.

Használt komponensek

Menu (LeftSideMenu.vue)

TableRow (GroupFilesRow.vue)

- Átadásra kerül az adott csoport id-je, ami alapján majd kilistázza a fájlokat. Átadunk még egy függvényit is nofilemessage nélkül, amit majd visszaküld ha nincsen fájl a csoportban, és kiírjuk hogy „A csoportban nem található fájl!”

Állapotok

data

store

- A bejelentkezett felhasználó lekéréséhez szükséges

groups

- Ebben lesznek a felhasználó csoportjainak adatai

nofilemessage

- Ez felel a „A csoportban nem található fájl” üzenet megjelenítéséért.

methods

loadgroups

- Ez a metódus tölti be a csoport adatait, majd adja át a *groups*-nak.

leavegroup

- Ez a metódus fut le, ha a felhasználó kilép a csoportból
- Paramétere az adott csoport id-je, melyet elküld a backend `api/group/${id}/leave` útvonalára.

UsersView

Ez a nézet 6 fő részből áll: Felhasználó kereső, Keresés eredménye, Barátok, Csoportok, Elküldött barát jelölések, Bejövő barát jelölések.

Ezen az oldalon lehet lebonyolítani a többi felhasználóval végzett lehetséges interakciókat és a csoportok létrehozását.

Felhasználó kereső

```
<form @submit.prevent="findUsers">
  <label for="keresesInput">Felhasználó keresése</label>
  <input type="search" name="" id="keresesInput" v-model="searchvalue" required />
  <button id="keresesGomb">Keresés</button>
</form>
<footer>
  <ul id="lettersearch">
    <li v-for="letter in letters" :key="letter"><a @click="searchbyletter(letter)">{{
      letter.toUpperCase()
    }}</a></li>
  </ul>
</footer>
```


Keresés eredménye

```


Találat a következő keresésre: {{ searchvalue }}



Nincs találat



- {{ user.nev }}

![add_friend_icon](@/assets/add_friend_icon.svg)

```

Barátok

```


# Barátok



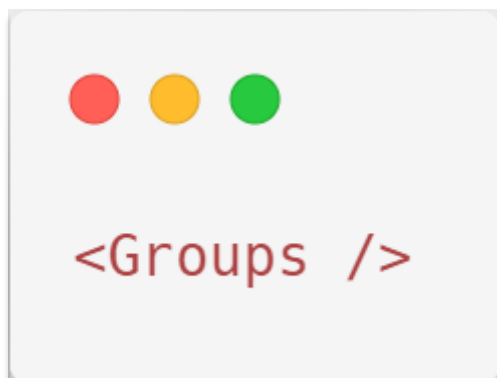
Jelenleg nincs egy barátod sem!



- {{ friend[0].nev }}
![add_friend_icon](@/assets/remove_friend_icon.svg)


```

Csoportok



Elküldött barát jelölések

```


# Függőben lévő jelöléseid



Nincs elküldött kérésed!



- {{ friend[0].nev }}



![add_friend_icon](@/assets/remove_friend_icon.svg)


```

Bejövő barát jelölések.

```


# Bejelöltek barátnak



Nincs függőben lévő jelölésed!



- {{ user[0].nev }}



![add_friend_icon](@/assets/accept_friend_icon.svg)


```

Használt komponensek

Menu (LeftSideMenu.vue)

Groups (Groups.vue)

- Ez a komponens jeleníti meg csoportok menüt. Ebben lehet létrehozni, törölni és feltölteni csoportokat.

Állapotok

data

users

- A felhasználó keresés eredményét tárolja.

friends

- A felhasználó barátait tárolja.

friendrequests

- A felhasználó bejövő barát kéréseit tárolja.

searchvalue

- A felhasználó kereső kulcsszavát tárolja.

sentrequests

- A felhasználó elküldött barát jelöléseit tárolja.

submitted

- Tárolja, hogy a keresés el lett-e indítva. Ez alapján jelenik meg a keresés eredménye mező.

store

- A bejelentkezett felhasználó lekéréséhez szükséges.

letters

- Az ABC betűi vannak benne. A kezdőbetűs kereséshez szükséges.

methods

getUsers

- A backend api/users útvonaláról betölti az összes felhasználót.

getFriends

- A backend api/user/friends útvonaláról betölti az aktuális felhasználó barátait.

getFriendRequests

- A backend api/user/friends/friendrequests útvonaláról betölti az aktuális felhasználó elküldött barát kéréseit.

findUsers

- A backend api/users/{username} útvonaláról betölti a felhasználó keresés eredményét.
- Paramétere a *searchvalue*

addfriend

- Elküldi a barát kérést a backend `api/user/friends/add` útvonalára.
- Paraméter a bejelölendő felhasználó id-je.

removefriend

- Elküldi a barát eltávolítása kérést a backend `api/user/friends/remove` útvonalára.
- Paraméter a törölendő felhasználó id-je.

acceptfriend

- Elküldi a barát elfogadása kérést a backend `api/user/friends/accept` útvonalára.
- Paraméter az elfogadandó jelölés id-je.

getSentrequests

- A backend `api/user/friends/sentrequests` útvonaláról betölti az adott felhasználó elküldött barát jelöléseit.

cancelrequest

- A backend `api/user/friends/cancelrequest` útvonalára elküldi a barátjelölés visszavonása kérést.
- Paraméter az visszavonása jelölés id-je.

reject

- A backend `api/user/friends/rejectrequest` útvonalára elküldi a barátjelölés elutasítása kérést.
- Paraméter az elutasítandó jelölés id-je.

searchbyletter

- Felhasználó keresés kezdőbetű alapján.
- A backend `api/users/letter/{startletter}` útfonalára küldi a kérést.

AdminView

```
<div id="adminpage">
  <div id="adminmenu">
    <button class="button" @click="changecomponent('files')">Fájlok kezelése</button>
    <button class="button" @click="changecomponent('users')">Felhasználók kezelése</button>
  </div>
  <div id="admincontent">
    <Files v-if="showfiles" />
    <Users v-if="showusers" />
  </div>
</div>
```

Ez a menüpont csak admin szerepkörű felhasználóknak jelenik- és tekinthető meg. Ha a szerepkör nem egyezik akkor a felhasználó visszairányításra kerül.

Használt komponensek

Menu (LeftSideMenu.vue)

Files (admin/Files.vue)

Users (admin/Users.vue)

Állapotok

data

showfiles

- Ez alapján jelenik meg a Files komponens.

showusers

- Ez alapján jelenik meg a Users komponens.

methods

changecomponent

- Paramétere a megjeleníteni kívánt komponens.
- A paraméter alapján beállítja a *showfiles* és a *showusers* értékeit.

SettingView

```
<div id="whiteBackground">
  <div id="left">
    <button id="dataButton" @click="displayDatas">Adatok</button>
    <button id="securityButton" @click="displaySecurity">Biztonság</button>
  </div>
  <div id="right">
    <ul v-if="errors != null" id="errorul">
      <li v-for="error in errors">
        <ul>
          <li v-for="err in error">{{ err }}</li>
        </ul>
      </li>
    </ul>
    <form method="post" @submit.prevent="shavechanges" enctype="multipart/form-data">
      <Datas id="picturegrid" v-if="showdata" :changedsettings="settings" />
      <Security v-if="showsecurity" :changedsettings="settings" />
      <div id="save">
        <input type="password" name="confirm" id="confirm" placeholder="Jelszó megerősítése" class="input"
          v-model="confirm" />
        <button class="button">Módosítások mentése</button>
      </div>
    </form>
  </div>
</div>
```

Használt komponensek

Menu (LeftSideMenu.vue)

Datas (settings/setDatas.vue)

A felhasználó adatainak beállításai – email cím, profil kép

Secutiry (settings/setSecurity.vue)

Biztonsági beállítások – jelszó, fiók törlése

Állapotok

data

showdata

- Ez alapján jelenik meg a Datas komponens.

showsecurity

- Ez alapján jelenik meg a Security komponens.

settings

- FormData típusú objektum. Ehhez lesznek hozzáadva a módosítandó beállítások.

errors

- A mentés gomb megnyomásakor, ha jön a backendről hibaüzenet, ebben lesz eltárolva.

confirm

- A jelszó megerősítése értékét tárolja.

store

- A bejelentkezett felhasználó azonosításához szükséges.

methods

displayDatas, displaySecurity

- Beállítják a showdata és a showsecurity változókat aszerint, melyik komponensnek kell megjelennie.

shavechanges

- A változtatott beállításokat elküldi a backend api/user/settings útvonalára.

Unauthorized



```
<h1 class="h1">Az oldal megtekintéséhez be kell jelentkezned!</h1>  
<h1 id="counter" v-text="counter"></h1>
```

Az oldal akkor jelenik meg, ha a felhasználó bejelentkezés nélkül akarja megtekinteni a weboldalt.

Állapotok

data

counter

- A visszaszámláló aktuális értékét tárolja

mounted

- 5-től kezdve számol visszafelé másodpercenként, és amit eléri a nullát, átdob a bejelentkezés oldalra.

LogoutPage



```
<h1 class="h1">Sikeres kijelentkezés</h1>  
<h1 id="counter" v-text="counter"></h1>
```

Az oldal akkor jelenik meg, ha a felhasználó a kijelentkezés gombra kattint, vagy a bejelentkezve töltött ideje elérte a 30 percet aktivitás nélkül.

Állapotok

data

counter

- A visszaszámláló aktuális értékét tárolja

mounted

- 5-től kezdve számol visszafelé másodpercenként, és amit eléri a nullát, átdob a bejelentkezés oldalra.

NotFoundPage



```
<h1 class="h1">Az oldal nem található</h1>
<div class="container">

  <div class="row">
    <div class="col loader">
      <span class="loader__element"></span>
      <span class="loader__element"></span>
      <span class="loader__element"></span>
    </div>
  </div>

  <div class="row">
    <div class="bgc">
      <h2 class="hero-title back-front tittle">
        404
      </h2>
    </div>
  </div>

  <div class="row">
    <div class="bgc">
      <p>Úgy látszik hogy találtál egy nem létező oldalt!</p>
      <p>Menj vissza az előző oldalra és próbáld meg újra!</p>
    </div>
  </div>

  <RouterLink to="/index" class="indexbutton">
    <span></span>
    <span></span>
    <span></span>
    <span></span>
    Tovább a főoldalra
  </RouterLink>
```

Az oldal akkor jelenik meg, ha a felhasználó olyan linket próbál elérni, amely nem létezik.

Nem tartalmaz javascript kódot, kizárólag html+css-ből épül fel.

Komponensek

Index

Az Index komponens a főoldal megjelenítéséért felelős. A tartalom része 3 elemből áll.

- 1) Legutóbbi fájlok
 - a. Találma 1 komponensből áll: LastFilesTable (LastFilesTable.vue)
 - b. Ez ugyan az a komponens, mint ami a *Saját fájlok* menüpontban található, azzal a különbséggel, hogy csak a 3 legutóbb módosított fájlt jeleníti meg.
 - c. A komponensnek a *lastfilesobject* paraméterébe átadásra kerül a *lastfiles* nevű objektumban tárolt adatok, valamint a *refresh* paraméterbe az adatok újratöltéséért felelős függvény.
- 2) Statisztika
 - a. Elemei *StatElement* komponensek. Megjelenítik hogy fájltípusonként (Zene, Videó, Kép, Dokumentum) és összesen mekkora méretben található a fájlok az adott felhasználói fiókban.
- 3) Fájl felöltése
 - a. Ebben a menüpontban lehet fájlt feltölteni az adatbázisba.
 - b. A fájlok feltöltését az *uploadfile* metódus végzi.
 - c. A form-ban az enctype biztosítja, hogy fájlokat lehessen küldeni.

Állapotok

data

lastfiles

- Ebbe a változóba fogja betölteni a *getlastfiles* metódus a 3 legutóbb módosított fájlt.

stat

- Ebbe a változóba tölti be a *getStat* metódus a statisztika adatait.

methods

uploadfile

- Egy FormData objektumba összegyűjti a Fájl feltöltése menüpont inputjából a betallózott fájlokat, majd a FormData objektummal együtt továbbítja a backendnek őket. Sikeres feltöltés esetén betöltődnek újra a legutóbbi fájlok és a statisztika.

getlastfiles

- A backend *api/file/get/{limit?}* útvonaláról betölti az adott legutóbbi 3 fájlját, majd átadja a *lastfiles* változónak.

getStat

- A backend *api/file/stat* útvonaláról betölti az adott felhasználó statisztikáját, majd átadja a kapott adatokat a *stat* változónak.

Login

A Login komponens a bejelentkező felület megjelenítéséért felelős. Itt van elhelyezve a bejelentkező form. Az input mezők a html adottságait kihasználva validálják a bemeneti adatokat. A bejelentkezés alatt lehetőség van jelszó emlékeztető email küldésére is.

Használt komponensek

LoginOrRegister (LoginOrRegister.vue)

- Ez a komponens 2 gombot tartalmaz, amelyek megjelenítik a kiválasztott felületet. (Bejelentkezés/Regisztráció)

Változók

data

- Ebben tárolódnak a bejelentkezéshez szükséges adatok. (felhasználónév-jelszó)

store

- Ebbe tárolódnak a bejelentkezett felhasználó adatai.

passwordreminder

- Ebbe van eltárolva az email cím, amire a jelszóemlékeztető el lesz küldve.

mailed

- Felelős az email sikeresen elküldve üzenet megjelenítéséért.

remindershow

- Felelős a jelszó emlékeztető mező megjelenítéséért.

Függvények

submit

- A bejelentkezés form küldés eseménye. Elküldi a backendnek a bekért felhasználónév-jelszó párost. Sikeres bejelentkezés esetén a store-ba eltárolja a bejelentkezett felhasználó adatait, majd átirányít a főoldalra.

displayreminder

- Megjeleníti a jelszó emlékeztető mezőt

reminder

- Elküldi a backendnek a jelszó emlékeztető kérést a megadott email címmel.

Register

A Register komponens a regisztrációs felület megjelenítéséért felelős. Itt van elhelyezve a regisztrációs form. Az input mezők a html adottságait kihasználva validálják a bemeneti adatokat. Amennyiben mégis hibás adatok kerülnének elküldésre, a backend visszaküldött hibaüzenetei az egyes input mezők alatt jelennek meg. Sikeres regisztráció esetén „Sikeres regisztráció” üzenetet jelenít meg, és 5 másodperc után automatikusan átirányít a bejelentkező felületre.

Használt komponensek

LoginOrRegister (LoginOrRegister.vue)

Állapotok

data

errors

- Ebbe a változóba fogja betöltődni a hibaüzenetek, ha a regisztrációs űrlap hibásan lenne elküldve.

data

- Ebbe az objektumban vannak tárolva a regisztrációs űrlap adatai.

methods

submit

- Ez a metódus küldi el a backendnek a regisztrációs űrlap adatait. Megnézi, hogy a jelszó megerősítése mező egyezik-e a megadott jelszóval. Amennyiben nem, erről tájékoztatja a felhasználót. Sikeres elküldés esetén átirányít a bejelentkező felületre.

LastFilesTable

Ez a tábla jeleníti meg a felhasználók saját fájljait. Lehetőséget biztosít több fájl egyszerre kijelölésére és rajta műveletek végrehajtására.

Használt komponensek

LastFilesTableRow (LastFilesTableRow.vue)

Állapotok

data

lastfiles

- Benne tárolódnak a megjelenítésre szánt fájlok. Tartalmát a szülő komponensétől kapja a lastfilesobject paraméteren keresztül. Tartalma automatikusan frissül a watch állapotban.

nofilemessage

- Ez a változó felelős a „Nincs feltöltött fájlod” üzenet megjelenítéséért.

checked

- Ez alapján lesz beállítva, hogy a táblázat összes eleme ki van-e jelölve vagy nincs kijelölve.

checkedBoxesData

- A kijelölt jelölőnégyzetek adatait tárolja. (Pontosabban az abban a sorban lévő fájl adatait)

store

- A bejelentkezett felhasználó eléréséhez szükséges.

groups

- A csoportos megosztáshoz tárolja a felhasználó csoportjainak adatait.

methods

getgroups

- A felhasználó csoportjainak adatait kéri el a backendtől.

checkall

- Ki- és bejelöli az összes táblázatban lévő fájlt.

download

- Letölt egy adott fájlt. A downloadselected metódusban van felhasználva.

deletefile

- Kitöröl egy adott fájlt. A deleteselected metódusban van felhasználva.

share

- Megoszt egy adott fájlt. A shareselected metódusban van felhasználva.

downloadselected

- Letölti a táblázatban kijelölt fájlokat.

deleteselected

- Kitörli a táblázatban kijelölt fájlokat.

shareselected

- Megosztja a táblázatban kijelölt fájlokat.

LastFilesTableRow

Ez a komponens adja a LastFilesTable komponens sorait. Megjeleníti az egyes fájlok adatait, és a lenyíló menüjében lehetőséget biztosít a fájlokat törölni, letölteni és megosztani.

LeftSideMenu

Ez a komponens adja az oldal menüjét. Az éppen használt menüpont gombját minden oldalon kijelöli. Megjelenik benne a profilkép, és egy számláló, amely az oldalon tétlenül biztosított 30 perctől kezdve számol vissza, és a végén kijelentkezik.

LogOutCounter

A LeftSideMenu komponensnél említett számlálót valósítja meg.

StatElement

A főoldali statisztika elemeinek komponens. Meg lehet adni neki milyen kategória nevet jelenítsen meg, az ikont hozzá, a foglalt tárhely méretét, az össz tárhely méretét, és hogy milyen színnel jelenjen meg.

ProgressBar

Mivel a HTML által biztosított progress elem nem volt megfelelő a célra, szükséges volt saját komponens létrehozása. Be lehet állítani a szélességét, a tárhely maximum méretét, a megtelt tárhely méretét és a kitöltött rész színét. A két méret adat és a maximum szélesség alapján százalékosan kiszámolja hogy mennyi rész legyen színes, és mennyi üres.

Groups

Ez a komponens a *Felhasználók* menüpontban van felhasználva. Itt található a csoportok létrehozása menü, és a létrehozott csoportok kilistázása, illetve tájékoztat, ha nincsen létrehozott csoportunk. Alkomponense a Groupmembers (Groupmembers.vue)

Groupmembers

Ez a komponens listázza ki az adott csoport tagjait. Lehetőség van a tagokat eltávolítani a csoportból.

GroupFilesRow

Ez a komponens a *Velem megosztva* aloldalon kapott helyet. Ebben vannak megjelenítve azon csoportban lévő fájlok adatai, amelyekhez hozzá van adva az adott felhasználó.

settings/setDatas

A beállítások Adatok menüpontja található benne. Ebben lehet kiválasztani/kitörölni a profilképet, és az email címet beállítani.

settings/setSecurity

A beállítások Biztonság menüpontja található benne. Itt lehet megváltoztatni a jelszót, és a felhasználói fiókot törölni. (A fiók törlés admin szerepkörrel nem lehetséges)

admin/Files

Az admin menü Fájlok kezelése menüpontja. Itt található egy teljes adatbázisra kiterjedő fájlkereső. Lehet keresni fájlnevre, és tulajdonos névre. Ezek kombinálhatóak is, illetve üres adatok esetén minden fájlt megjelenít.

admin/Users

Az admin menü Felhasználók kezelése menüpont. Található benne egy felhasználó kereső, ahol a felhasználó nevére / nevének részletére lehet keresni. Itt lehet felhasználókat törölni, és nekik jogosultságot adni.

Android alkalmazás

A FiveFile-hoz készült Androidon futtatható alkalmazás is. Itt kizárólag korlátozott funkciók érhetőek el, hiszen a fő profil a webalkalmazás lefejlesztése volt. Ezek a funkciók a feltöltött fájlok kilistázása, letöltése, törlése, beállításokban email cím és jelszó megváltoztatása, valamint a felhasználói fiók törlése. Az alkalmazás Flutter-rel készült. Ennek előnye, hogy a kód mind webre, Androidra, és IOS-re is lefordítható.

Az alkalmazásban 4 útvonal – 4 menüpont található.

Az útvonalak definiálása a *main.dart* fájlban történt.

1. /login: Bejelentkező oldal (pages/login.dart)
2. /register: Regisztrációs oldal (pages/register.dart)
3. /home: Főoldal (pages/home.dart)
4. /files: Saját fájlok (pages/files.dart)
5. /settings: Beállítások (pages/settings.dart)
6. /profil: Profil (pages/profil.dart)

login.dart

Az oldalon elhelyezett komponens osztályok:

- LoginTitle: az oldal címét jeleníti meg egy Text widgetben.
- UserName: A felhasználónév beviteli mezője. Tartalma bekerül a *username* kontrollerbe.
- Password: A jelszó beviteli mezője. Tartalma bekerül a *password* kontrollerbe.
- Bejelentkezés gomb: A szükséges adatok kitöltése esetén meghívja a *UserController login* metódusát, paraméterben átadva a felhasználónév-jelszó párost.
- Regisztráció gomb: Átírnyít a /register útvonalra.

register.dart

Az oldalon elhelyezett komponens osztályok:

- RegisterTitle: az oldal címét jeleníti meg egy Text widgetben.
- UserName: A felhasználónév beviteli mezője. Tartalma bekerül a *username* kontrollerbe.
- Email: Az email cím beviteli mezője. Tartalma bekerül az *email* kontrollerbe.
- Password: A jelszó beviteli mezője. Tartalma bekerül az *password* kontrollerbe.
- PasswordConfirm: A jelszó megerősítés beviteli mezője. Tartalma bekerül az *passwordConfirm* kontrollerbe.
- Regisztrálok gomb: a fenti kontrollerek adatait elküldi a *UserController register* metódusának, majd átírnyít a /login útvonalra.

home.dart

- Ez az oldal tájékoztató jellegű. Közli a felhasználóval hogy korlátozott funkciókkal elérhető az app.

files.dart

- A backendet futtatva minden esetben látni, hogy egy töltő képernyő várakoztatja a felhasználót, amíg az adatai betöltődnek. Ha nincsen mit betölteni akkor kiírjuk a felhasználónak, hogy nincsenek feltöltött fájljai. Mindez a FutureBuilder felhasználásával történt. A FutureBuildernek a future attribútumába megadtuk a *UserController loadFiles* metódusát, amely betölti a felhasználó fájljait. Amit megérkeztek az adatok, kilistázásra kerülnek. A kilistázás *ListView widgetbe* történt, ezen belül mindegyik elem be van ágyazva egy *Slidable widgetbe*. A Slidable widget *startActionPane* attribútumában van megadva, hogy milyen effekttel történjen a widget mozgatása, és hogy milyen widgetek jelenjenek meg elmozgatás esetén. DeleteFileButton és DownloadButton. Ezek a children attribútumban vannak elhelyezve. Mindkét gombnak átadásra kerül a backendtől várt fájl név, amely szerint tudja meg törölni, illetve letölteni az adott fájlokat. Az oldal frissíthetőségéről a RefreshIndicator widget gondoskodik.

settings.dart

Az oldalon elhelyezett komponens osztályok:

- Email: Az email cím beviteli mezője. Tartalma bekerül az *email* kontrollerbe. A hinttext attribútumában megjelenik a bejelentkezett felhasználó aktuális email címe.
- Password: A jelszó beviteli mezője. Tartalma bekerül az *password* kontrollerbe.
- Confirm: A jelszó megerősítés beviteli mezője. Tartalma bekerül a *confirm* kontrollerbe.
- Mentés gomb: a fenti kontrollerek adatait elküldi a *UserController settings* metódusának.

profil.dart

Az oldalon elhelyezett komponensek:

- Köszöntő mező: Üdvözlí a belépett felhasználót.
- Felhasználónév mező: Megjeleníti a belépett felhasználó felhasználónevét.
- Email cím mező: Megjeleníti a belépett felhasználó email címét.
- Regisztráció dátuma mező: Megjeleníti mikor regisztrált a felhasználó.
- Profil törlés jelölőnégyzet: Ha be van pipálva, kijelentkezéskor törlődik a felhasználói profil.
- Kijelentkezés gomb: Kijelentkezik az oldalról.

file_model.dart

- A felhasználó fájljainak modelje. Tárolja a fájlról az azonosítóját, a tulajdonos azonosítóját, típusát, nevét, méretét, a megosztott csoport azonosítóját, feltöltés és módosítás dátumát.

file_model.g.dart

- Generált fájl.
- A beérkező fájlok átalakítását végzi.

urlprefix.dart

- A backend requestek egyszerű átírását könnyíti meg.

user_model.dart

- A bejelentkezett felhasználóról tárolja az azonosítóját, nevét, email címét, szerepkörét, profilképét és regisztrációjának dátumát.

user_controller.dart

Az alábbi kéréseket küldi a backend felé:

- login: Bejelentkezés
- register: Regisztráció
- logout: Kijelentkezés
- settings: Beállítások mentése
- loadFiles: Fájlok betöltése
- deleteFile: Kiválasztott fájl törlése

menu.dart

- Ebben a fájlban található az oldal menü komponense.

DeleteFileButton.dart

- Ebben a fájlban található a kiválasztott fájl törléséért felelős gomb. Paramétere a törlendő fájl neve, és azt tovább küldi a *UserController deleteFile* metódusának.

DeleteFileButton.dart

- Ebben a fájlban található a kiválasztott fájl letöltéséért felelős gomb. A letöltött fájl a Letöltések mappába fog kerülni.

A programozáshoz használt források jegyzéke

1. HTML: HyperText Markup Language
 - a. <https://developer.mozilla.org/en-US/docs/Web/HTML>
2. CSS: Cascading Style Sheets
 - a. <https://developer.mozilla.org/en-US/docs/Web/CSS>
3. JavaScript
 - a. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
4. W3Schools
 - a. <https://www.w3schools.com/>
5. Laravel
 - a. <https://laravel.com/docs/8.x/installation>
6. Laravel Collective
 - a. <https://laravelcollective.com/docs/6.x/html>
7. Vue JS
 - a. <https://vuejs.org/>
8. Pinia
 - a. <https://pinia.vuejs.org/>
9. Flutter
 - a. <https://flutter.dev/>
10. Stack Overflow
 - a. <https://stackoverflow.com/>