

CollaboraCode

Fejlesztői dokumentáció

2023

Fejlesztők

Borbély Gergő Árpád (mobil alkalmazás)

Ferencsik Boglárka (frontend)

Nagy-Berta Lázár (backend)

Tartalomjegyzék

A CollaboraCode célja	3
Felhasznált technológiák	3
Követelmények	4
Funkcionális	4
Nem funkcionális	4
Tervek	5
Frontend	7
Komponensek leírása	7
Mobil alkalmazás	13
Komponensek leírása	13
Service-ek leírása	23
Guard leírása	28
Pipe leírása	29
Backend	30
Végpontok	30
Users	30
Projects	31
Files	33
Kontrollerek	35
UsersController	35
ProjectsController	39
FilesController	44
Modellek	49
File	49
Project	49
Project_member	50
User	50
Tárgymutató	51
Ábrajegyzék	55
Felhasznált Irodalom	56
Köszönetnyilvánítás	57

A CollaboraCode célja

A CollaboraCode azért jött létre, hogy segítse az olyan programozók munkáját, akik csapatban dolgoznak együtt más fejlesztőkkel. A CollaboraCode-ot használva közösen együttműködve tudnak kódot írni, és azt a beépített futtató környezet segítségével ki is tudják próbálni.

Felhasznált technológiák

Frontend:

- Angular CLI 17.1.1
- Tailwind CSS 3.4.1
- CodeMirror 5
- Judge0 és RapidApi (kódfuttató környezet, Pro csomag előfizetés)
- Vercel (frontend hosting)

Mobil alkalmazás:

- Angular CLI - 17.0.6
- Tailwind CSS - 3.4.1
- Ionic - 7.2.0

Backend:

- AdonisJS v5
- MySQL 8.0.30
- Aiven (database hosting)
- Render (backend hosting)

Fejlesztési módszer:

- Kanban (Trello)

Követelmények

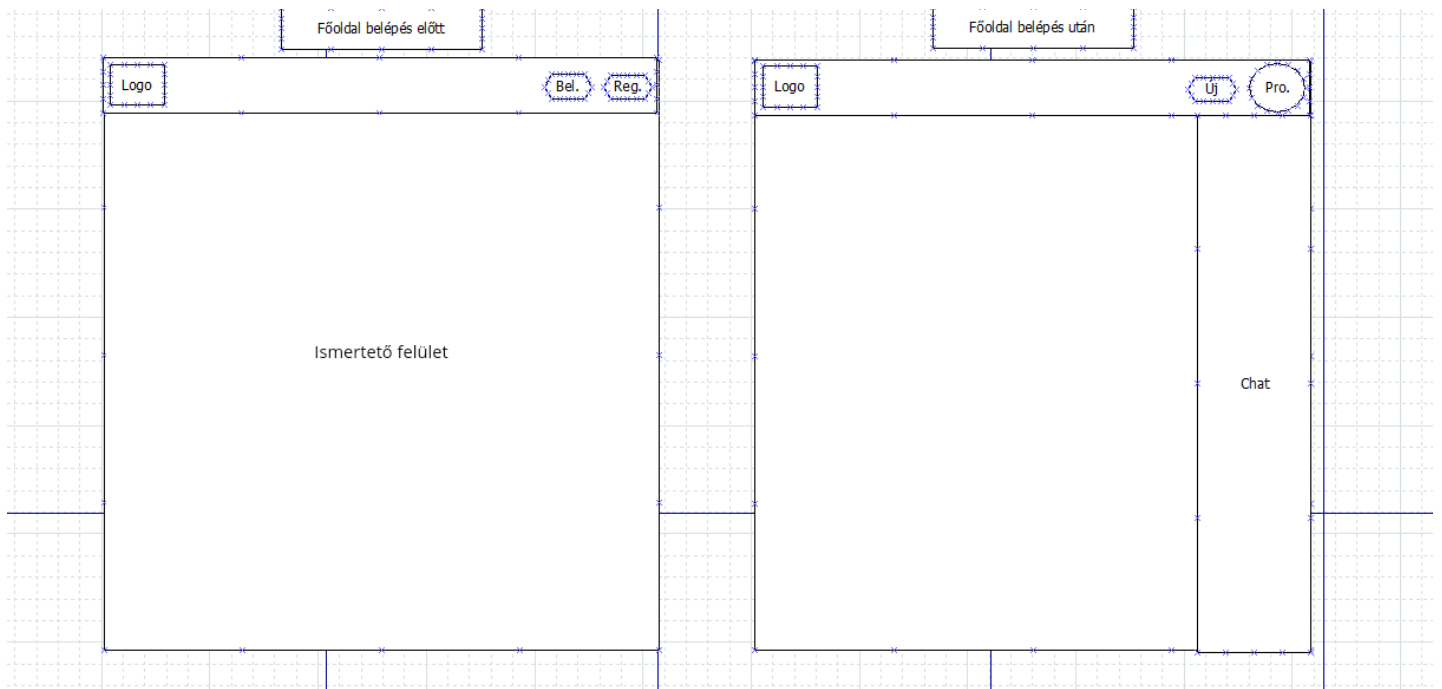
Funkcionális

- Az oldal minden funkciójának elérése regisztrációhoz kötött.
- Regisztrációkor szükséges a vezetéknév, keresztnév, e-mail cím és jelszó megadása.
- Regisztrációkor a jelszónak tartalmaznia kell minimum 8, maximum 16 karaktert, legalább 1 számot, illetve kis- és nagybetűket.
- Belépni az e-mail cím és jelszó párossal lehetséges.
- Regisztráció/belépés után van lehetőség projektek létrehozására.
- Meglévő projektbe lehet fájlt készíteni.
- A fájlt megnyitva van lehetőség annak tartalmának szerkesztésére.
- Van lehetőség a profil adatok megváltoztatására.

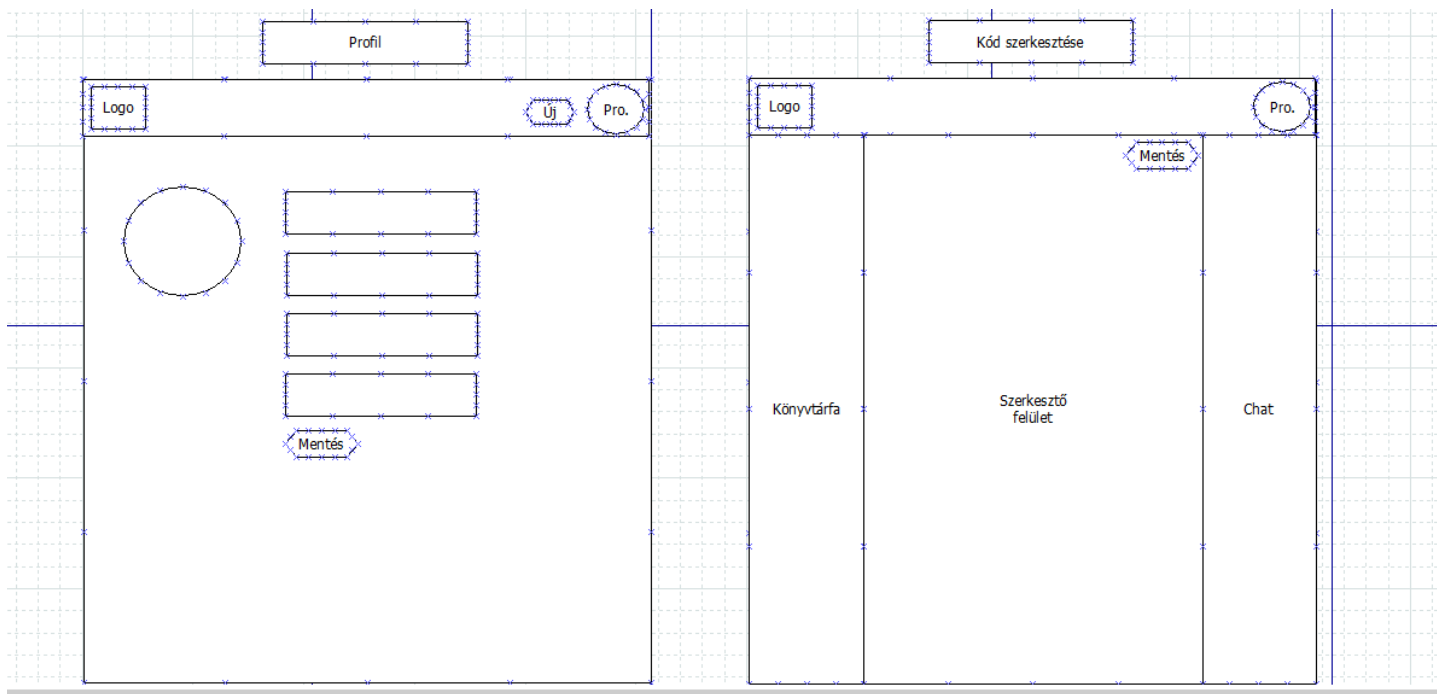
Nem funkcionális

- A frontend Angular 17.1.1 keretrendszerrel készült és a Vercel hostolja.
- A kódfuttató környezet a RapidApi kiszolgálón hostolt Judge0 segítségével működik, aminek a Pro csomagjára van előfizetve egy hónapra az optimális használat érdekében.
- Az adatbázis MySQL alapú és az Aiven nevű kiszolgálón keresztül érhető el.
- A backend AdonisJS v5 keretrendszerrel készült és a Render nevű kiszolgálón van hostolva.
- A mobil alkalmazás Angular 17.0.6 és Ionic 7.2.0 keretrendszerekkel készült

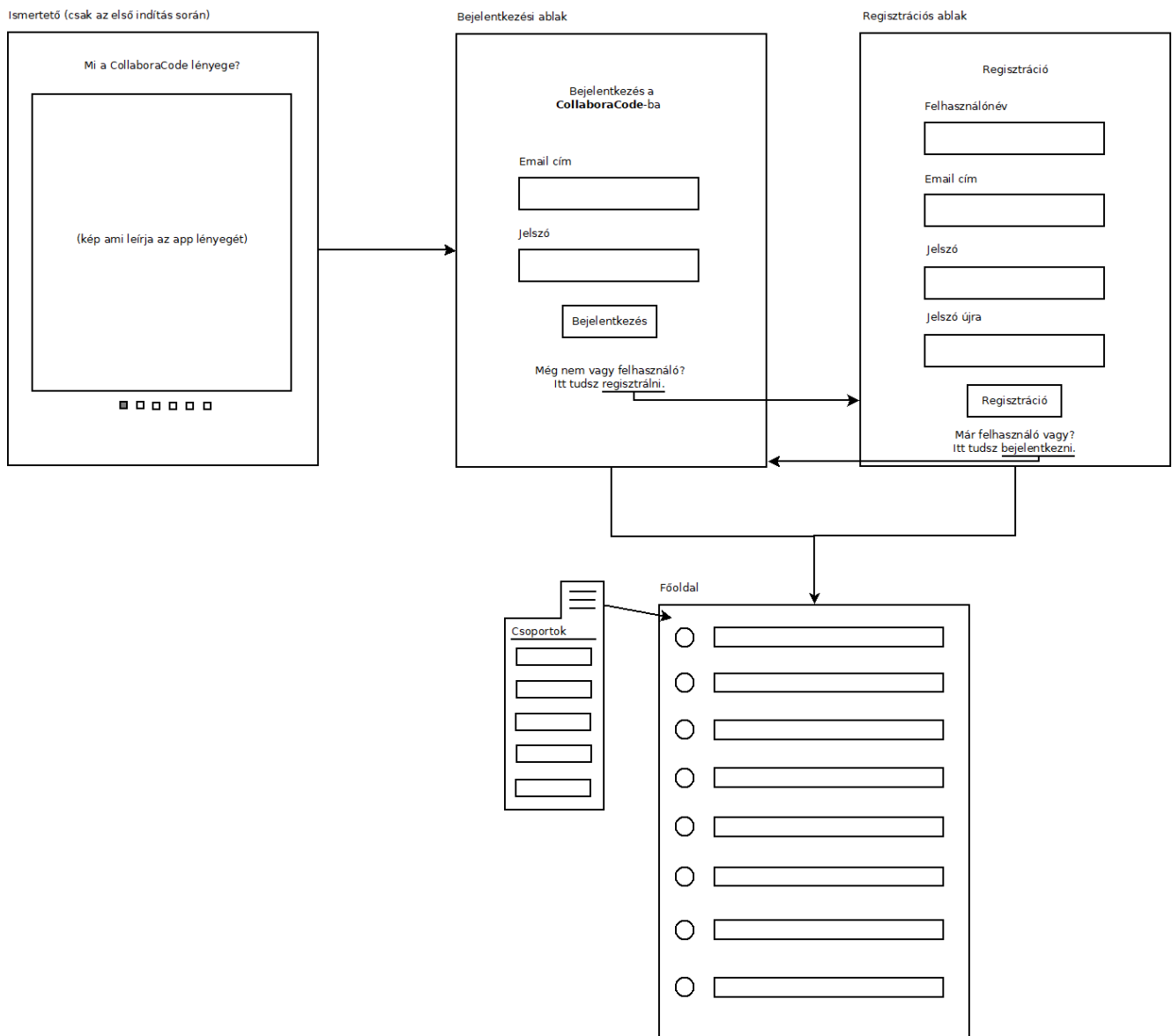
Tervek



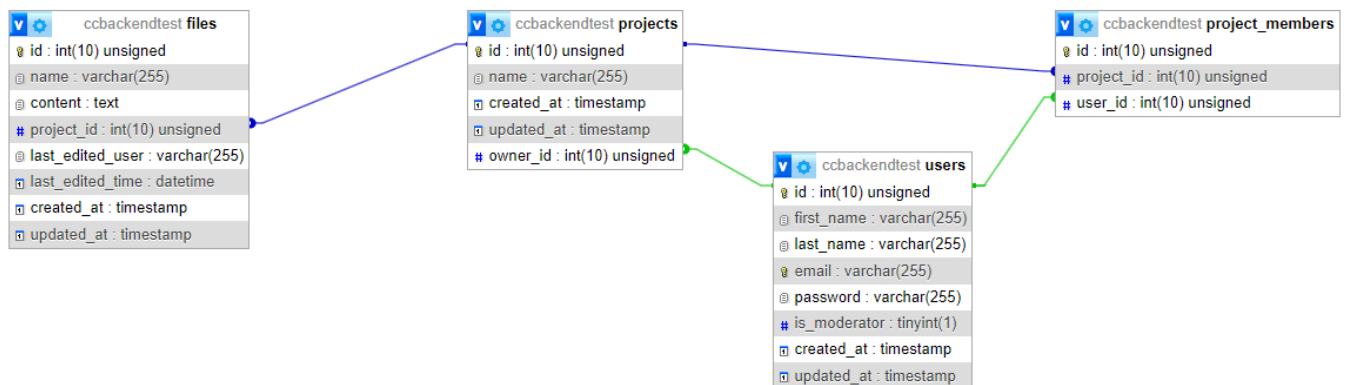
1.1 ábra: Webes felület kezdetleges terve



1.2 ábra: Webes felület kezdetleges terve



2. ábra: Mobilos felület kezdetleges terve



3. ábra: Adatbázis terv

Frontend

Komponensek leírása

- **Add-people:**

- A projekt neve alapján küld egy POST kérést a project/info-nak, ami továbbítja a project_id-t és a megadott e-mail címet a project/member/add végpontra és hozzárendelni a felhasználót a projekthez.

- **Editor:**

- Az oldal betöltésekor küld egy POST kérést a file/info végpontra, ami a fájl azonosítóját elküldi a file/content végpontra és az megjeleníti a fájl tartalmát. A futtatás gomb elküldi a kiválasztott programozási nyelv azonosítóját és a szerkesztő tartalmát a Judge0 online futtató környezetének, majd az az output mezőben megjeleníti a lefuttatott kódot. A mentés gomb pedig szintén POST kérést küld a file/info végpontra, és a fájl neve alapján továbbítja a fájl azonosítót és a fájl tartalmát a file/edit végpontra, és egy PUT kéréssel módosítja a tartalmat.

- **Files:**

- Az oldal betöltésekor küld egy POST kérést a megadott projekt nevével a project/info végpontra, ami továbbítja a projekt azonosítóját egy POST kéréssel a project/files végpontra, ami lekéri az adott projektben lévő fájlokat. Az új fájl gomb egy POST kéréssel továbbítja az adott projekt nevét a project/info végpontra, ami pedig elküldi a projekt azonosítóját, illetve a megadott fájl nevet egy POST kéréssel a file/create végpontra, ami létrehozza az új fájlt. A projekt törlése gomb pedig először egy POST kéréssel lekérdezi a projekt azonosítóját a project/info végponton, majd azt továbbítva egy DELETE kérést küld a project/delete végpontra, ami törli a projektet és siker esetén visszairányítja a felhasználót a projects komponens oldalára.

- **Footer:**

- Egy egyszerű footer, rajta a "CollaboraCode" felirattal, egy copyright jellel és az éppen aktuális évvel. Az editor komponens oldalán kívül minden oldal alján látható.

- **Home:**
 - Ide vannak behivatkozva a nav, a home-before és a footer komponensek, ezt látja legelőször a felhasználó, aki megnyitja az oldalt.
- **Home-before:**
 - Belépés/regisztráció előtt ez az oldal látható. Az oldal tetején látható a navigációs sáv, az alatt pedig egy üdvözlő üzenet egy egyszerű ábrával, ami demonstrálja az oldal működését.
- **Login:**
 - POST kérést küld a login végpontra a felhasználó e-mail címével és jelszavával, és sikeres autentikáció esetén átirányítja a projects komponens oldalára.
- **Nav:**
 - A belépés/regisztráció előtti főoldal tetején elhelyezkedő navigációs sáv. Megtalálható rajta a kis logó, ami egy home gombként funkcionál, a teljes logó, illetve a belépés és a regisztráció gombok, amik átirányítanak a megfelelő komponens oldalára.
- **Nav-after:**
 - A belépés/regisztráció utáni oldalak tetején elhelyezkedő navigációs sáv. Megtalálható rajta a kis logó, ami egy home gombként funkcionál, a teljes logó, illetve egy lenyíló menü, ami tartalmazza a profil és a projects komponensek felületére átirányító gombokat, illetve egy kijelentkezés gombot, ami POST kérést küld a logout végpontra, kijelentkezteti a felhasználót, és siker esetén visszairányítja a home-before komponens oldalára.
- **Profil:**
 - Az oldal betöltésekor küld egy GET kérést az account/info végpontra, ami feltölti a vezetéknev és keresztnév input mezőket a felhasználó adataival. A módosítás gomb pedig PUT kérést küld az account/modify végpontra a megadott adatokkal. A fiók törlése gomb DELETE kérést küld, és sikeres törlés esetén visszairányítja a felhasználót a home-before komponens oldalára.

- **Projects:**

- Az oldal betöltésekor küld egy GET kérést az account/info végpontra, ami továbbítja az e-mail címet egy POST kéréssel a user/projects végpont felé és megjeleníti mind a saját, mind a megosztott projekteket. Az új projekt gomb a megjelenő input mező tartalmával küld egy POST kérést a project/create végpontra, ami létrehozza az új projektet a megadott névvel.

- **Sign-up:**

- POST kérést küld a sign-up végpontra a felhasználó vezetéknévvel, keresztnévvel, e-mail címével és jelszavával, amit szükséges megadni megerősítés végett még egyszer. Sikeres regisztráció esetén átirányít a projects komponens oldalára.

- **Auth service:**

- login: Meghívja az authenticationClient login metódusát, ami továbbküldi a kérelmet a szerver felé. Sikeres bejelentkezés esetén a szerver a tokenet küldi vissza válaszban, majd a böngésző helyi tárolójába menti és ezután átirányítja a felhasználót a projects komponens oldalára. Ellenkező esetben pedig megjelenít egy felugró ablakot, tájékoztatva a felhasználót a hibáról.
- signUp: Meghívja az authenticationClient register metódusát. A kérelem tartalmazza a first_name (string), last_name (string), email (string), password (string) és password_confirmation (string) adatokat. Ha a regisztráció sikeres, a szerver a tokenet küldi vissza, majd a böngésző helyi tárolójába menti és ezután megjelenít egy felugró ablakot, ami tájékoztatja a felhasználót a sikeres regisztrációról, aztán tovább irányítja a projects komponens oldalára. Sikertelen regisztráció esetén megjelenít egy felugró ablakot, ami tájékoztatja a felhasználót a hibáról.
- logout: A localStorage removeItem metódusával eltávolítja a tokenet a böngésző helyi tárolójából, majd visszairányítja a felhasználót a kezdőlapra.
- isLoggedIn: Ellenőrzi, hogy a felhasználó be van-e jelentkezve, ha igen, beolvassa a helyi tárolóban rögzített tokenet. Visszatérési értéke true vagy false lehet.

- getToken: Meghívja az isLoggedIn metódust, hogy ellenőrizze a felhasználó be van-e jelentkezve. Ha igen, akkor beolvassa a token-t és visszaküldi string formátumban. Ellenkező esetben null értékkel tér vissza.
- Backend service:
 - getHeaders: Egy HTTP headers objektumot ad vissza, ami tartalmazza azt, hogy a kérés JSON formátumú, illetve a token-t, amivel azonosítani lehet az adott felhasználót.
 - createProject: Új projekt létrehozásáért felelős metódus. Lekéri a projekt nevét és a token-t, majd küld egy POST kérést a project/create végpontra. A visszatérő válasz siker esetén a létrehozott projekt adatait tartalmazza, hiba esetén pedig hibaüzenetet ad vissza.
 - createFile: Ez a metódus új fájl létrehozásáért felelős. Lekéri a fájl és a projekt nevét, illetve a token-t. Azután küld egy POST kérést a projekt nevével a project/info végpontra, majd ha a projekt létezik, egy POST kérést küld a file/create végpontra a fájl nevével és a projekt azonosítójával. Ha a projekt nem létezik, hibaüzenetet ad vissza.
 - openFile: Egy adott fájlt nyit meg. Lekéri a fájl nevét és a token-t. POST kérést küld a file/info végpontra a fájl nevével. Ha a fájl létezik, akkor POST kéréssel elküldi a fájl azonosítóját a file/content végpontra. Siker esetén visszaadja a fájl tartalmát, ellenkező esetben hibaüzenetet ad vissza.
 - editCode: Egy adott fájl tartalmát módosítja. Lekéri a fájl nevét, tartalmát és a token-t. Küld egy POST kérést a file/info végpontra, majd ha a fájl létezik, PUT kéréssel elküldi a fájl azonosítóját és a fájl módosított tartalmát. Ellenkező esetben hibaüzenetet ad vissza.
 - addUser: Hozzáad egy új szerkesztőt a megadott projekthez. Lekéri a szerkesztő e-mail címét, a projekt nevét és a token-t. Ezután POST kéréssel elküldi a projekt nevét a project/info végpontra. Ha a projekt létezik, akkor POST kérést küld a project/member/add végpontra és elküldi az e-mail címet és a projektazonosítót. Ellenkező esetben hibaüzenetet ad vissza.

- `getAccountInfo`: Lekéri a bejelentkezett felhasználó fiókjának adatait. Lekéri a token, majd egy GET kérést küld az `account/info` végpontra, ami visszaadja az adatokat.
- `deleteProject`: Töröl egy adott projektet. Lekéri a projekt nevét és a token. POST kérést küld a `project/info` végpontra és elküldi a projekt nevét. Ezután DELETE kérést küld a `project/delete` végpontra a projekt azonosítójával. Siker esetén üres observable jön vissza, ellenkező esetben hibaüzenetet ad vissza.
- `getProjectInfo`: Lekéri egy adott projekt adatait. Lekéri a projekt nevét és a token, majd POST kérést küld a `project/info` végpontra és elküldi a projektnevet. Ha a projekt létezik, visszaadja a projekt adatait JSON formátumban. Ellenkező esetben hibaüzenetet dob.
- `modifyUser`: Módosítja a bejelentkezett felhasználó adatait. Lekéri a módosítandó adatokat (vezetéknév, keresztnév, jelszó) és a token. Ezután PUT kérést küld az `account/modify` végpontra a módosítandó adatokkal. Sikeres módosítás esetén üres objektumot ad vissza, ellenkező esetben hibaüzenetet ad.
- `deleteAccount`: Törli a felhasználó fiókját. Lekéri a token, majd DELETE kérést küld az `account/delete` végpontra. Sikeres törlés esetén üres objektumot ad vissza, ellenkező esetben pedig hibaüzenetet ad vissza.
- `getUserProjects`: Lekéri a felhasználó saját és megosztott projektjeit. Lekéri a token, majd GET kérést küld az `account/info` végpontra. Ha a fiók létezik, lekéri az e-mail címet. Ezután POST kérést küld a `user/projects` végpontra és elküldi a felhasználó e-mail címét. Sikeres lekérés esetén a projekteket adja vissza, ellenkező esetben hibaüzenetet küld.
- **ProjectService:**
 - `getProjectNameFromUrl`: Az adott projekt nevét nyeri ki az aktuális URL-ből. A `window.location.pathname` tulajdonsággal lekérdezi az aktuális URL-t. Split metódussal felosztja az URL-t listaelemekre, majd az utolsó elemet adja vissza, mint a projekt neve.

- `getFileNameFromUrl`: Az adott fájl nevét nyeri ki az aktuális URL-ből. A `window.location.pathname` tulajdonsággal lekérdezi az aktuális URL-t. `Split` metódussal felosztja az URL-t listaelemekre, majd az utolsó elemet adja vissza, mint a fájl neve.
- `CompileService`:
 - `createSubmission`: Létrehoz egy `data` objektumot, ami tartalmazza a programozási nyelv azonosítóját és a futtatni kívánt nyelv forráskódját. Létrehoz egy `headers` objektumot, ami meghatározza hogy az elküldött adat JSON típusú, és elküldi a RapidAPI szolgáltatáshoz titkos kulcsát, illetve a Judge0 API hostnevét. Küld egy POST kérést a `hostnév/submissions` végpontra visszatérési értéke pedig a Judge0 API válasza.
 - `getSubmission`: Létrehoz egy timer-t, ami 1 másodpercenként futtatja a megadott callback függvényt. A callback függvény létrehoz egy URL-t a token paraméter alapján a Judge0 API lekérdezési végpontjához. Létrehoz egy `headers` objektumot a RapidAPI kulccsal és a hostnévvel. Ezután GET kérést küld a létrehozott URL-re. Ha a válasz sikeres, és a válasz objektumának státusz azonosítója nem egyenlő 1-gyel, akkor leállítja az időzítőt és lekéri a Judge0 válaszát. Ha a kérés során hiba történik, akkor leállítja az időzítőt és hibaüzenetet ad vissza.

Mobil alkalmazás

Komponensek leírása

- **about**

- Leírás
 - Ez a komponens megjeleníti az applikációhoz felhasznált technológiákat, a projekt fejlesztőit és az app verziószámát.
- URL
 - /tabs/settings/about
- Bemenő paraméterek
 - Nincsen bemenő paraméter
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **account-settings**

- Leírás
 - Ez a komponens a fiók adatok módosítására szolgál. Ezen a lapon lehet megváltoztatni a felhasználó keresztnévét, vezetéknévét és a fiók jelszavát.
- URL
 - /tabs/settings/account
- Bemenő paraméterek
 - Meghívott függvény
 - baseService - getAccountInformation
 - Kapott értékek
 - Keresztnév
 - Vezetéknév
 - Email-cím (nincsen felhasználva a komponensben)
- Kimenő paraméterek
 - Meghívott függvény
 - baseService - updateAccountInformation
 - Küldött értékek
 - Keresztnév
 - Vezetéknév
 - Jelszó

- **create-file**

- Leírás

- Ez a komponens egy fájl létrehozására szolgál. Meg lehet adni a fájl nevét és a gombra kattintva átvizsgál a **project-details** komponensre. A bal felső sarokban lévő vissza gombra kattintva pedig a **projects-overview** komponens tölt be.

- URL

- /tabs/projects/create-file

- Bemenő paraméterek

- Nincsen bemenő paraméter

- Kimenő paraméterek

- Meghívott függvény
 - baseService - createFile
 - Küldött értékek
 - Fájlnev

- **create-project**

- Leírás

- Ez a komponens egy projekt létrehozására szolgál. Meg lehet adni a projekt nevét és a gombra kattintva átvizsgál a **project-details** komponensre. A bal felső sarokban lévő vissza gombra kattintva a **projects-overview** tölt be.

- URL

- /tabs/projects/create-file

- Bemenő paraméterek

- Nincsen bemenő paraméter

- Kimenő paraméterek

- Meghívott függvény
 - baseService - createProject
 - Küldött értékek
 - Projektnév

- **error**

- Leírás
 - Ez a komponens akkor tölt be, ha a megadott url nem található. A lap 5 másodperc után átirányít a **home** komponensre, vagy manuálisan is át lehet lépni a kártyán lévő link segítségével.
- URL
 - /error vagy hibás url esetén a megadott url
- Bemenő paraméterek
 - Nincsen bemenő paraméter
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **file-card**

- Leírás
 - Ez a komponens egy fájl-nak az utolsó módosítási adatait jeleníti meg. A kártyára való kattintással a **show-file-changes** komponens tölt be.
- URL
 - /tabs/projects/file-card
- Bemenő paraméterek
 - Kapott érték:
 - Bemenet
 - Fájl
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **file-settings**

- Leírás
 - Ez a komponens a fájl adatainak módosítására szolgál. Itt lehet módosítani a fájl nevét és törölni a fájlt is.
- URL
 - /tabs/projects/file-settings
- Bemenő paraméterek
 - Meghívott függvény
 - baseService - getFileInfoWithName
 - Kapott értékek
 - Fájlnev
- Kimenő paraméterek
 - Meghívott függvény
 - baseService - updateFile
 - baseService - deleteFile
 - Küldött értékek
 - updateFile
 - Fájlazonosító
 - Fájlnev
 - deleteFile
 - Fájlazonosító

- **home**

- Leírás
 - Ez a komponens a főoldal-t jeleníti meg. A főoldalon lesznek majd láthatóak a legutolsó módosítások.
- URL
 - /tabs/home
- Bemenő paraméterek
 - Nincsen bemenő paraméter
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **login**

- Leírás
 - Ez a komponens a bejelentkezési oldalt tartalmazza. A bejelentkezéshez az email cím és a jelszó szükséges. A gombra kattintva a **home** komponens tölt be.
- URL
 - /login
- Bemenő paraméterek
 - Nincsen bemenő paraméter
- Kimenő paraméterek
 - Meghívott függvény
 - authService - login
 - Küldött értékek
 - Email cím
 - Jelszó

- **project-card**

- Leírás
 - Ez a komponens egy kártya ami a szerkesztők adatait jeleníti meg, aki a megadott projektbe fel lett véve.
- URL
 - /tabs/projects/details/:name
- Bemenő paraméterek
 - Meghívott függvény
 - baseService - getProjectInfoWithId
 - Kapott értékek
 - Bemenet
 - Fájlnév
 - getProjectInfoWithId
 - Projekt
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **project-details**

- Leírás
 - Ez a komponens megjeleníti az elérhető fájlokat. a fájl kártyára kattintva a **file-card** komponens jön be. A jobb felső sarokban lévő beállítások ikonra kattintva a **project-settings** tölt be
- URL
 - /tabs/projects/details/:name
- Bemenő paraméterek
 - Meghívott függvények
 - baseService - getProjectInfoWithName
 - baseService - getProjectFiles
 - baseService - getLastEditInformation
 - Kapott értékek
 - getProjectInfoWithName
 - Projektazonosító
 - getProjectFiles
 - Projekt fájljai
 - getLastEditInformation
 - Utolsó szerkesztési információ
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **project-settings**

- Leírás
 - Ez a komponens a projekt adatainak módosítására szolgál. Itt lehet módosítani a projekt nevét és a projektet törölni. Projekt törlése esetén a **project-details** komponens tölt be.
- URL
 - /tabs/projects/project-settings
- Bemenő paraméterek
 - Nincsen bemenő paraméter
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **projects**

- Leírás
 - Ez a komponens a projekttel kapcsolatos komponenseket tölti be. Csak útvonal kezelés céljából szükséges.
- URL
 - /tabs/projects
- Bemenő paraméterek
 - Nincsen bemenő paraméter
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **projects-overview**

- Leírás
 - Ez a komponens a saját és a megosztott projekteket kérdezi le és jeleníti meg.
- URL
 - /tabs/projects
- Bemenő paraméterek
 - Meghívott függvény
 - baseService - getUserProjects
 - Kapott érték
 - Projekt
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **register**

- Leírás
 - Ez a komponens egy felhasználói fiók regisztrációjához szükséges. A regisztrációhoz szükséges a felhasználó keresztnéve, vezetéknéve, email címe és a jelszava. A jelszót meg kell erősíteni. A regisztrációra kattintva a **home** komponens töltődik be.
- URL
 - /register
- Bemenő paraméterek
 - Nincsen bemenő paraméter
- Kimenő paraméterek
 - Meghívott függvények
 - authService - register
 - Küldött értékek
 - Keresztnév
 - Vezetéknév
 - Email cím
 - Jelszó
 - Jelszó újra

- **settings**

- Leírás
 - Ez a komponens a beállításokkal kapcsolatos komponenseket tölti be. Csak útvonal kezelés céljából szükséges.
- URL
 - /tabs/settings
- Bemenő paraméterek
 - Nincsen bemenő paraméter
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **settings-options**

- Leírás
 - Ez a komponens a beállítási menüket jeleníti meg. Az egyik menü kártyájára kattintva az adott komponens tölt be. Az alsó kijelentkezés gomb segítségével a **login** komponens tölt be.
- URL
 - /tabs/settings
- Bemenő paraméterek
 - Nincsen bemenő paraméter
- Kimenő paraméterek
 - Meghívott függvény
 - authService - logout
 - Küldött érték
 - Nincsen küldött érték

- **show-file-changes**

- Leírás
 - Ez a komponens a kiválasztott kártyáról bővebb információt jelenít meg. A jobb felső sarokban lévő beállítások ikonra kattintva a **file-settings** komponens tölt be, a bal felső vissza gombra kattintva pedig a **project-details** tölt be.
- URL
 - /tabs/projects/details/:name/
- Bemenő paraméterek
 - Kapott értékek
 - Bemenet
 - Utolsó szerkesztési információ
 - Jelenlegi projekt név
 - File változásait követő objektum állapota
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **tabs**

- Leírás
 - Ez a komponens az alul lévő navigációs sáv megjelenítésére szolgál. Az ikonokra kattintva az adott oldal jelenik meg.
- URL
 - /tabs
- Bemenő paraméterek
 - Meghívott függvények
 - authService - renewToken
 - Kapott érték
 - Nincs kapott érték
- Kimenő paraméterek
 - Nincsen kimenő paraméter

- **theme-settings**

- Leírás
 - Ez a komponens a téma beállítására szolgál. a bal felső sarokban lévő vissza kombra kattintva a **settings** komponens tölt be
- URL
 - /tabs/settings/theme
- Bemenő paraméterek
 - Nincsen bemenő paraméter
- Kimenő paraméterek
 - Nincsen kimenő paraméter

Service-ek leírása

- **auth**

- Metódusok

- isAuthenticated

- Leírás

- Ellenőrzi, hogy a felhasználó be van-e jelentkezve a tárolt token alapján.

- Visszatérési érték

- boolean

- setToken

- Leírás

- Beállítja a megadott token értékét a localStorage-ban.

- Küldött adatok

- string (token)

- getToken

- Leírás

- Lekéri a felhasználói tokent a localStorage-ból.

- Visszatérési érték

- string

- register

- Leírás

- Új felhasználót regisztrál a megadott adatokkal. A válasz az egy HTTP válasz objektum, mely tartalmazza a válasz státuszát és a szerver válaszüzenetét.

- Meghívott végpont

- /sign-up

- Kérés típusa

- POST

- Küldött adatok

- Felhasználó regisztrációs adatok (firstName, lastName, email, password, passwordConfirmation)

- Visszatérési érték

- Observable<HttpResponse<any>>

■ login

- Leírás
 - Bejelentkezteti a felhasználót a megadott adatokkal. A válasz az egy HTTP válasz objektum, mely tartalmazza a válasz státuszát, a token-t és egyéb információkat a szerverről.
- Meghívott végpont
 - /login
- Kérés típusa
 - POST
- Küldött adatok
 - Bejelentkezési adatok (email, password)
- Visszatérési érték
 - Observable<HttpResponse<any>>

■ logout

- Leírás
 - Kijelentkezteti a felhasználót. A token törlésre kerül a localStorage-ből. Opcionálisan átirányítja a felhasználót a **/login** oldalra.
- Meghívott végpont
 - /logout
- Kérés típusa
 - POST
- Visszatérési érték
 - Observable<HttpResponse<any>>

■ renewToken

- Leírás
 - Megújítja a lejárfélben lévő token. A megújítás sikeressége esetén frissíti a localStorage-ban tárolt token az új értékre.
- Meghívott végpont
 - /token/renew
- Kérés típusa
 - POST

- **base**

- Metódusok

- getPasswordIcon

- Leírás
 - Visszaadja a jelszó ikonjának elérési útvonalát.
 - Bemeneti érték
 - passwordVisibility: boolean
 - Visszatérési érték
 - string

- getAccountInformation

- Leírás
 - Lekéri a felhasználó fiókinformációit a token alapján.
 - Meghívott végpont
 - account/info
 - Visszatérési érték
 - Observable<any>

- updateAccountInformation

- Leírás
 - Frissíti a felhasználó fiókinformációit.
 - Meghívott végpont
 - account/modify
 - Bemeneti érték
 - firstName: string, lastName: string, password: string
 - Visszatérési érték
 - Observable<any>

- deleteAccount

- Leírás
 - Törli a felhasználó fiókját.
 - Meghívott végpont
 - account/delete
 - Visszatérési érték
 - Observable<any>

- getProjectInfoWithId

- Leírás
 - Lekéri a projektinformációkat az azonosító alapján.
 - Meghívott végpont
 - project/info
 - Bemeneti érték
 - id: number
 - Visszatérési érték
 - Observable<any>

- **getProjectInfoWithName**
 - Leírás
 - Lekéri a projektinformációkat a név alapján.
 - Meghívott végpont
 - project/info
 - Bemeneti érték
 - name: string
 - Visszatérési érték
 - Observable<any>
- **getUserProjects**
 - Leírás
 - Lekéri a felhasználóhoz tartozó projekteket.
 - Meghívott végpont
 - user/projects
 - Visszatérési érték
 - Observable<any>
- **createProject**
 - Leírás
 - Létrehoz egy új projektet.
 - Meghívott végpont
 - project/create
 - Bemeneti érték
 - projectName: string, users: any[]
 - Visszatérési érték
 - Observable<any>
- **getProjectFiles**
 - Leírás
 - Lekéri a projekt fájljait.
 - Meghívott végpont
 - project/files
 - Visszatérési érték
 - Observable<any[]>
- **getFileInfoWithId**
 - Leírás
 - Lekéri a fájl információit az azonosító alapján.
 - Meghívott végpont
 - file/info
 - Bemeneti érték
 - id: number
 - Visszatérési érték
 - Observable<any>

- **getFileInfoWithId**
 - **Leírás**
 - Lekéri a fájl információit a név alapján.
 - **Meghívott végpont**
 - file/info
 - **Bemeneti érték**
 - name: string
 - **Visszatérési érték**
 - Observable<any>
- **createFile**
 - **Leírás**
 - Létrehoz egy új fájlt a projektben.
 - **Meghívott végpont**
 - file/create
 - **Bemeneti érték**
 - fileName: string
 - **Visszatérési érték**
 - Observable<any>
- **updateFile**
 - **Leírás**
 - Frissíti a fájl információit.
 - **Meghívott végpont**
 - file/modify
 - **Bemeneti érték**
 - id: number, fileName: string
 - **Visszatérési érték**
 - Observable<any>
- **deleteFile**
 - **Leírás**
 - Törli a fájlt a projektből.
 - **Meghívott végpont**
 - file/delete
 - **Bemeneti érték**
 - id: number
 - **Visszatérési érték**
 - Observable<any>

- **updateProject**
 - **Leírás**
 - Frissíti a projekt információit.
 - **Meghívott végpont**
 - project/modify
 - **Bemeneti érték**
 - id: number, projectName: string
 - **Visszatérési érték**
 - Observable<any>
- **deleteProject**
 - **Leírás**
 - Törli a projektet.
 - **Meghívott végpont**
 - project/delete
 - **Bemeneti érték**
 - id: number
 - **Visszatérési érték**
 - Observable<any>
- **getLastEditInformation**
 - **Leírás**
 - Lekéri a legutóbbi szerkesztés információit.
 - **Meghívott végpont**
 - file/info/edit
 - **Bemeneti érték**
 - fileId: number
 - **Visszatérési érték**
 - Observable<any>

Guard leírása

- **auth**

- **Leírás**

- Az *AuthGuard* egy Angular guard, amely biztosítja, hogy csak az autentikált felhasználók férjenek hozzá bizonyos útvonalakhoz az alkalmazásban.
- Az *AuthGuard* funkciója az, hogy ellenőrzi, hogy a felhasználó autentikált-e vagy sem. Ha a felhasználó autentikált, engedélyezi az útvonalhoz való hozzáférést, ellenkező esetben pedig átirányítja a felhasználót a bejelentkezési oldalra.
- Az *AuthGuard*-ot a routing modulban kell regisztrálni azokon az útvonalakon, amelyekhez autentikációhoz van szükség.
- Ez azt jelenti, hogy az **/tabs** útvonalhoz csak az autentikált felhasználók férhetnek hozzá.

Pipe leírása

- **hungarian-date**

- Leírás

- A HungarianDatePipe egy Angular csővezeték (pipe), amely dátumokat formáz magyar lokalizációval.

- Metódusok

- transform(value: any): any
 - Ez a metódus a pipe fő metódusa, amely a bejövő értéket átalakítja a kívánt formátumba.

- Paraméterek

- value: A bejövő dátum érték, amelyet formázni kell.

- Visszatérési érték

- A formázott dátum string formában.

- Használat

- A HungarianDatePipe-et a transform metódus segítségével alkalmazzuk a bejövő dátum értékre. A moment segítségével a dátumot magyar lokalizációra állítja (locale('hu')), majd a megadott formátumban formázza (format('YYYY. MM. DD - HH:mm')). Ezután visszaadja a formázott dátumot.
 - {{ someDate | hungarianDate }}
 - Ez például a 2024-04-17T12:30:00 dátumot formázza a következő formátumba: 2024. 04. 17 - 12:30.

- Függőségek

- moment: Egy nagyon népszerű JavaScript dátumkezelő könyvtár.

Backend

Végpontok

Users

- **account/info:**
 - Típus: GET
 - Bejövő paraméter: token (string)
 - Művelet: fiók adatainak lekérése
 - Visszatérési érték: id (integer), firstName (string), lastName (string), email (string)
- **sign-up:**
 - Típus: POST
 - Bejövő paraméterek: last_name (string), first_name (string), email (string), password (string), password_confirmation (string)
 - Művelet: végrehajtja a regisztrációs folyamatot
 - Visszatérési érték: token (string)
- **login:**
 - Típus: POST
 - Bejövő paraméterek: email (string), password (string)
 - Művelet: végrehajtja a bejelentkezési folyamatot
 - Visszatérési érték: token (string)
- **logout:**
 - Típus: POST
 - Bejövő paraméter: token (string)
 - Művelet: kijelentkezteti a felhasználót
 - Visszatérési érték: -
- **verify:**
 - Típus: POST
 - Bejövő paraméter: token (string)
 - Művelet: lekéri a felhasználó tokenjének lejáratási idejét
 - Visszatérési érték: dátum (dateTime)

- **token/renew:**

- Típus: POST
- Bejövő paraméter: token (string)
- Tevékenység: lecseréli a felhasználó előző tokenét
- Visszatérési érték: új token (string)

- **account/modify:**

- Típus: PUT
- Bejövő paraméterek: last_name (string), first_name (string), password (string), token (string)
- Művelet: módosítja a felhasználó adatait
- Visszatérési érték: last_name (string), first_name (string), password (string)

- **account/delete:**

- Típus: DELETE
- Bejövő paraméter: token (string)
- Művelet: törli a felhasználó fiókját
- Visszatérési érték: -

Projects

- **project/info:**

- Típus: POST
- Bejövő paraméterek: name (string), token (string)
- Művelet: lekéri egy adott projekt adatait
- Visszatérési érték: id (integer), name (string), owner adatai: id (integer), firstName (string), lastName (string), email (string); members (array): id (integer), firstName (string), lastName (string), email (string)

- **user/projects:**
 - Típus: POST
 - Bejövő paraméterek: user_email (string), token (string)
 - Művelet: lekéri egy adott felhasználó projektjeit
 - Visszatérési érték: owns (array): id (integer), name (string); memberOf (array): id (integer), name (string), owner: id (integer), firstName (string), lastName (string), email (string);

- **project/create:**
 - Típus: POST
 - Bejövő paraméterek: name (string), token (string)
 - Művelet: létrehoz egy új projektet
 - Visszatérési érték: name (string), ownerId (integer), id (integer)

- **project/modify:**
 - Típus: PUT
 - Bejövő paraméterek: id (integer), name (string), token (string)
 - Művelet: egy adott projekt adatait módosítja
 - Visszatérési érték: name (string)

- **project/delete:**
 - Típus: DELETE
 - Bejövő paraméterek: id (integer), token (string)
 - Művelet: töröl egy adott projektet a projektazonosító alapján
 - Visszatérési érték: -

- **project/members:**
 - Típus: POST
 - Bejövő paraméterek: project_id (integer), token (string)
 - Művelet: lekéri egy adott projekt tagjait
 - Visszatérési érték: array: id (integer), firstName (string), lastName (string), email (string)

- **project/member/add:**

- Típus: POST
- Bejövő paraméterek: project_id (integer), user_email (string), token (string)
- Művelet: hozzáad egy felhasználót egy adott projekthez
- Visszatérési érték: -

- **project/member/remove:**

- Típus: DELETE
- Bejövő paraméterek: project_id (integer), user_email (string), token (string)
- Művelet: eltávolít egy felhasználót egy adott projektből
- Visszatérési érték: -

Files

- **file/info:**

- Típus: POST
- Bejövő paraméterek: id (integer), token (string)
- Művelet: lekéri egy adott fájl adatait
- Visszatérési érték: id (integer), name (string), projects adatai: id (integer), name (string)

- **project/files:**

- Típus: POST
- Bejövő paraméterek: project_id (integer), token (string)
- Művelet: lekéri egy adott projekt fájljait
- Visszatérési érték: array: id (integer), name (string)

- **file/create:**

- Típus: POST
- Bejövő paraméterek: name (string), project_id (integer), token (string)
- Művelet: létrehoz egy új fájlt, egy adott projekten belül
- Visszatérési érték: id (integer), name (string)

- **file/modify:**
 - Típus: PUT
 - Bejövő paraméterek: id (integer), name (string), token (string)
 - Művelet: módosítja egy adott fájl adatait
 - Visszatérési érték: name (string)
- **file/delete:**
 - Típus: DELETE
 - Bejövő paraméterek: id (integer), token (string)
 - Művelet: töröl egy fájlt, egy adott projekten belül
 - Visszatérési érték: -
- **file/content:**
 - Típus: POST
 - Bejövő paraméterek: id (integer), token (string)
 - Művelet: lekéri egy fájl tartalmát
 - Visszatérési érték: content (string)
- **file/edit:**
 - Típus: PUT
 - Bejövő paraméterek: id (integer), content (string)
 - Művelet: módosítja egy fájl tartalmát
 - Visszatérési érték: content (string)
- **file/info/edit:**
 - Típus: POST
 - Bejövő paraméterek: id (integer), token (string)
 - Művelet: lekéri egy adott fájl utolsó módosítását
 - Visszatérési érték: id (integer), name (string), lastEditedUser (string), lastEditedTime (dateTime)

Kontrollerek

UserController

- **getAccountInfo:**

- Megszólított végpont: account/info
- Bejövő paraméter: token (string)
- Visszatérési érték: id (integer), firstName (string), lastName (string) és email (string)
- Leírás: Lekéri az adott felhasználó adatait
- Működés: A User modellt használja az adatok lekéréséhez. Meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nincs, egy 401-es hibakódot ad vissza. Ha be van lépve, a bejövő paramétert a getUserInfoValidator ellenőrzi. Ezután lekérdezi a felhasználó adatait és 200-as státuszkóddal JSON formátumban küldi vissza az adatokat. Hiba esetén pedig 422-es hibakóddal tér vissza.

- **signUp:**

- Megszólított végpont: sign-up
- Bejövő paraméter: first_name (string), last_name (string), email (string), password (string), password confirmation (string)
- Visszatérési érték: token (string)
- Leírás: Végrehajtja a regisztrációs folyamatot
- Működés: A User modellt használja új felhasználók regisztrálására. Ezután a bejövő paramétereket a registerValidator ellenőrzi. Ha a validálás sikertelen 422-es hibakóddal tér vissza. Érvényes adatok esetén létrehozza a felhasználót és a token, majd küld a megadott e-mail címre egy tájékoztató üzenetet a regisztráció sikerességéről és 201-es státuszkóddal tér vissza.

- **login:**

- Megszólított végpont: login
- Bejövő paraméter: email (string), password (string)
- Visszatérési érték: token (string)
- Leírás: Bejelentkezteti a felhasználót
- Működés: A User modellt használja a felhasználó bejelentkezésének kezelésére. A bejövő paramétereket a loginValidator ellenőrzi. Érvénytelen adatok esetén 422-es hibakódot küld vissza. Érvényes adatok esetén a metódus kikeresi a felhasználót az e-mail címe alapján. Ha nem létezik a felhasználó, 400-as hibakóddal tér vissza. Ha a felhasználó létezik, a megadott jelszót összehasonlítja a tárol jelszóval. Egyezés esetén létrehoz egy tokent, aminek érvényessége 30 nap, és 200-as státuszkóddal JSON formátumban küldi vissza a választ. Ellenkező esetben 400-as hibakódot küld vissza.

- **verify:**

- Megszólított végpont: verify
- Bejövő paraméter: token (string)
- Visszatérési érték: dátum (dateTime)
- Leírás: Visszaküldi a felhasználó tokenjének lejáratát idejét.
- Működés: A User modellt használja a bejelentkezett felhasználó azonosságának ellenőrzésére. Először az authenticateUser metódust hívja meg, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nincs bejelentkezve, 401-es hibakóddal tér vissza, ellenkező esetben 200-as státuszkódot, és a token lejáratát idejét küldi vissza.

- **renewToken:**

- Megszólított végpont: token/renew
- Bejövő paraméter: token (string)
- Visszatérési érték: token (string)
- Leírás: Új tokent generál a felhasználónak
- Működés: Először az authenticateUser metódust hívja meg, és ellenőrzi a felhasználó be van-e lépve. Ha nincs, 401-es hibakóddal tér vissza, ellenkező esetben törli a bejövő paraméterként érkező tokent és létrehoz egy újat. Ha a csere sikeres, 200-as státuszkóddal tér vissza.

- **logout:**

- Megszólított végpont: logout
- Bejövő paraméter: token (string)
- Visszatérési érték: -
- Leírás: Kijelentkezteti a felhasználót
- Működés: A User modellt használja a felhasználói adatok kezelésére. Meghívja az authenticateUser metódust, és ellenőrzi, hogy a felhasználó be van-e lépve. Ha nincs, 401-es hibakóddal tér vissza. Ellenkező esetben pedig törli a tokent és 200-as státuszkóddal válaszol.

- **modifyAccount:**

- Megszólított végpont: account/modify
- Bejövő paraméter: last_name (string), first_name (string), password (string), token (string)
- Visszatérési érték: last_name (string), first_name (string), password (string)
- Leírás: A felhasználó adatait módosítja
- Működés: Elsőként az authenticateUser metódust hívja meg, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakóddal válaszol. Ellenkező esetben beolvassa a bejövő paraméterként érkező adatokat és a modifyAccountValidatorral való ellenőrzés után módosítja azokat. Sikeres módosítás esetén 200-as státuszkódot küld vissza, validálási hiba esetén pedig 422-es hibakóddal válaszol.

- **deleteAccount:**

- Megszólított végpont: account/delete
- Bejövő paraméter: token (string)
- Visszatérési érték: -
- Leírás: Törli a felhasználó fiókját
- Működés: Meghívja az authenticateUser metódust és ellenőrzi a felhasználó be van-e lépve. Ha nincs, 401-es hibakóddal tér vissza. Ellenkező esetben pedig a bejövő paraméterben érkező token alapján törli a felhasználó fiókját, ami nem visszafordítható művelet, és 200-as státuszkóddal válaszol.

- **authenticateUser:**

- Megszólított végpont: -
- Bejövő paraméter: -
- Visszatérési érték: -
- Leírás: Ellenőrzi, hogy a felhasználó be van-e lépve.

ProjectsController

- **getProjectInfo:**

- Megszólított végpont: project/info
- Bejövő paraméter: name (string), token (string)
- Visszatérési érték: id (integer), name (string), owner adatai: id (integer), firstName (string), lastName (string), email (string); members (array): id (integer), firstName (string), lastName (string), email (string)
- Leírás: Meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve, ha igen, lekéri a megadott projekt összes adatát a fájlakon kívül
- Működés:

- **getUserProjects:**

- Megszólított végpont: user/projects
- Bejövő paraméter: user_email (string), token (string)
- Visszatérési érték: Sikeres lekérés esetén visszaküldi a következő listákat: owns (array): id (integer), name (string); memberOf (array): id (integer), name (string), owner: id (integer), firstName (string), lastName (string), email (string);
- Működés: A Project és a ProjectMember modelleket használja az adatok lekéréséhez. Először az authenticateUser metódust hívja meg, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ezután beolvassa a bejövő paramétert és a getUserProjectsValidator validátorral ellenőrzi az adatot. A bejövő paraméter alapján lekéri a felhasználót az adatbázisból, majd lekéri a saját és megosztott projekteket. Végül sikeres lekérdezés esetén JSON formátumban küldi vissza az adatokat egy 200-as státuszkóddal. Hiba esetében 422-es hibakóddal tér vissza.

- **createProject:**

- Megszólított végpont: project/create
- Bejövő paraméter: name (string), token (string)
- Visszatérési érték: Hiba nélküli lefutás esetén visszaküldi a következő adatokat: name (string), ownerId (integer), id (integer).
- Leírás: Létrehoz egy új projektet
- Működés: Először meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nincs, 401-es hibakódot küld vissza. Ellenkező esetben a paraméterként beérkező névvel létrehoz egy objektumot, ami tartalmazza a projekt nevét és a tulajdonos azonosítóját. Ezeket az adatokat a createProjectValidator ellenőrzi. Sikeres validálás után 201-es státuszkóddal létrehozza a projektet. Hiba esetén 422-es hibakódot küld vissza.

- **modifyProject:**

- Megszólított végpont: project/modify
- Bejövő paraméter: id (integer), name (string), token (string)
- Visszatérési érték: name (string)
- Leírás: Meglévő projekt adatait módosítja
- Működés: Először meghívja az authenticateUser metódust és ellenőrzi a felhasználó be van-e jelentkezve. Az érkező kérésből kinyeri az adatokat, majd ezeket a modifyProjectValidator ellenőrzi. Ezt követően kikeresi a megfelelő projektet az adatbázisban, és ellenőrzi, hogy a kérést küldő felhasználó a projekt tulajdonosa-e. Ha nem, 401-es hibakóddal tér vissza. Ellenkező esetben frissíti a projekt nevét és 201-es státuszkódot küld vissza.

- **deleteProject:**

- Megszólított végpont: project/delete
- Bejövő paraméter: id (integer), token (string)
- Visszatérési érték: -
- Leírás: Töröl egy adott projektet
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. Ellenkező esetben az érkező kérésből kinyeri az adatokat és a deleteProjectValidator ellenőrzi azokat. Sikeres ellenőrzés esetén megkeresi a törölni kívánt projektet. Ezután ellenőrzi, hogy a kérést küldő felhasználó a projekt tulajdonosa-e. Ha nem, akkor a visszaküldött hibakód: 401. Sikeres törlés esetén a visszaküldött státuszkód: 200.

- **getMembers:**

- Megszólított végpont: project/members
- Bejövő paraméter: project_id (integer), token (string)
- Visszatérési érték: id (integer), firstName (string), lastName (string), email (string)
- Leírás: Lekéri az adott projekt tagjait
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. Ellenkező esetben a beérkező projektazonosítót a getMembersValidator ellenőrzi. Ezután ellenőrzi, hogy a kérést küldő felhasználó a projekt tagja-e. Ha nem, akkor 401-es hibakódot küld. Ha tagja a projektnek, akkor lekérdezi a szerkesztők adatait a projekt tulajdonosa kivételével, és 201-es státuszkóddal tér vissza.

- **addMember:**

- Megszólított végpont: project/member/add
- Bejövő paraméter: project_id (integer), user_email (string), token (string)
- Visszatérési érték: -
- Leírás: Hozzáad egy tagot egy adott projekthez
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. Ha be van lépve, a beérkező adatokat az addMemberValidator ellenőrzi. Ezután a projektazonosító alapján kikeresi az adott projektet, és a megadott e-mail cím alapján pedig kikeresi az adott felhasználót. Nem létező felhasználó esetén 401-es hibakóddal tér vissza. Ellenkező esetben ellenőrzi, hogy a kérést küldő felhasználó tagja-e a projektnek, ha nem, 401-es hibakódot küld vissza. Ezután vizsgálja, hogy a megadott felhasználó tagja-e már a projektnek vagy ő-e a projekt tulajdonosa. Mindkét esetben a visszaküldött hibakód: 401. Ha a megadott felhasználó nem tulajdonosa és nem tagja a projektnek, akkor hozzáadja a projekthez, és lekéri a hozzáadott tag adatait. Sikeres hozzáadás esetén 201-es státuszkód mellett JSON formátumban küldi el a tag adatait. Hiba esetén 401-es hibakódot küld.

- **removeMember:**

- Megszólított végpont: project/member/remove
- Bejövő paraméter: project_id (integer), user_email (string), token (string)
- Visszatérési érték: -
- Leírás: Eltávolít egy tagot egy adott projektből
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. Ellenkező esetben a beérkező adatokat a removeMemberValidator ellenőrzi az adatokat. A megadott projektazonosító alapján megkeresi a hozzátartozó projektet, majd a megadott e-mail cím alapján pedig kikeresi a hozzátartozó felhasználót. Ha a felhasználó nem létezik, vagy nem a projekt tagja, 401-es hibakóddal tér vissza. Ezután ellenőrzi, hogy az eltávolítani kívánt felhasználó a projekt tulajdonosa-e. Ha igen, szintén 401-es hibakódot küld. Ha a tag nem tulajdonos és tagja a projektnek, akkor eltávolítja a projektből. Sikeres eltávolítás esetén visszaküldi az eltávolított felhasználó adatait és 201-es státuszkódot küld vissza. Hiba esetén az elküldött hibakód: 401.

- **authenticateUser:**

- Megszólított végpont: -
- Bejövő paraméter: -
- Visszatérési érték: -
- Leírás: Ellenőrzi, hogy a felhasználó be van-e lépve

FilesController

- **getFileInfo:**

- Megszólított végpont: file/info
- Bejövő paraméter: id (integer), token (string)
- Visszatérési érték: id (integer), name (string), projects adatai: id (integer), name (string)
- Leírás: Lekéri egy adott fájl információit
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. Ha be van lépve, a beérkező adatokat a getFileInfoValidator ellenőrzi. A beérkező fájlazonosító alapján lekéri a fájl és a kapcsolódó projekt adatait. Sikeres lekérés esetén JSON formátumban visszaküldi a fájl adatait 200-as státuszkód mellett. Hiba esetén a hibakód: 422.

- **getProjectFiles:**

- Megszólított végpont: project/files
- Bejövő paraméter: project_id (integer), token (string)
- Visszatérési érték: id (integer), name (string)
- Leírás: Lekéri egy adott projekt összes fájlját
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. A bejövő paramétereket a getProjectFilesValidator ellenőrzi. A projektazonosító alapján megkeresi a projektet, majd lekérdezi a hozzátartozó fájlokat. A lekérdezés csak az azonosítót és a nevet küldi vissza JSON formátumban 200-as státuszkóddal. Hiba esetén 422-es hibakódot küld.

- **createFile:**

- Megszólított végpont: file/create
- Bejövő paraméter: name (string), project_id (integer), token (string)
- Visszatérési érték: id (integer), name (string).
- Leírás: Létrehoz egy fájlt egy adott projekten belül
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. Ha be van lépve, a bejövő paramétereket a createFileValidator ellenőrzi. A kérésből kiolvassa a bejelentkezett felhasználó nevét, amit hozzáfűz egy last_edited_user attribútumhoz. A last_edited_time az utolsó szerkesztés idejét határozza meg. Sikeres létrehozás esetén az adatok 201-es státuszkóddal JSON formátumban érkeznek. Hiba esetén 422-es hibakóddal tér vissza.

- **modifyFile:**

- Megszólított végpont: file/modify
- Bejövő paraméter: id (integer), name (string), token (string)
- Visszatérési érték: name (string)
- Leírás: Megváltoztatja egy adott fájl nevét
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. A beérkező paramétereket a modifyFileValidator ellenőrzi. Ezután kikeresi a módosítandó fájlt az adatbázisban, majd módosítja az adott fájl nevét. Sikeres módosítás esetén JSON formátumban visszaküldi a fájl adatait és 201-es státuszkódot küld vissza. Hiba esetén 422-es hibakódot küld.

- **deleteFile:**

- Megszólított végpont: file/delete
- Bejövő paraméter: id (integer), token (string)
- Visszatérési érték: -
- Leírás: Töröl egy adott fájlt egy adott projektből
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. A bejövő paramétert a deleteFileValidator ellenőrzi. A megadott projektazonosító alapján kikeresi a törölni kívánt fájlt az adatbázisban és lekéri az adatait. Sikeres törlés esetén 200-as státuszkód mellett JSON formátumban küldi vissza a törölt fájl adatait.

- **getFileContent:**

- Megszólított végpont: file/content
- Bejövő paraméter: id (integer), token (string)
- Visszatérési érték: content (string)
- Leírás: Lekéri egy adott fájl tartalmát
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. A beérkező paramétert a getFileContentValidator ellenőrzi. A megadott azonosító alapján lekérdezi az adott fájl tartalmát és a hozzá kapcsolódó projekt azonosítóját. Ellenőrzi, hogy a felhasználó tagja-e az adott projektnek. Ha nem, 403-as hibakóddal tér vissza. Ellenkező esetben 200-as státuszkóddal tér vissza a fájl tartalmával együtt.

- **editFileContent:**

- Megszólított végpont: file/edit
- Bejövő paraméter: id (integer), content (string), token (string)
- Visszatérési érték: content (string)
- Leírás: Módosítja egy adott fájl tartalmát
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. A bejövő paramétereket az editFileContentValidator ellenőrzi. Ellenőrzi, hogy a kérést küldő felhasználó tagja-e a projektnek, amihez a fájl tartozik. Ha nem tagja, akkor 403-as hibakódot küld vissza. Ha tagja, akkor a fájl utolsó szerkesztésének dátumát és szerkesztőjét frissíti a kérést küldő felhasználó adataival. Sikeres módosítás esetén visszaküldi a módosított fájl tartalmát 201-es státuszkód mellett. Hiba esetén 422-es hibakóddal tér vissza.

- **getLastEditInfo:**

- Megszólított végpont: file/info/edit
- Bejövő paraméter: id (integer), token (string)
- Visszatérési érték: id (integer), name (string), lastEditedUser (string), lastEditedTime (dateTime)
- Leírás: Lekéri egy adott fájl utolsó szerkesztési információit
- Működés: Elsőként meghívja az authenticateUser metódust, ami ellenőrzi, hogy a felhasználó be van-e lépve. Ha nem, 401-es hibakódot küld vissza. A bejövő paramétereket a getLastEditedInfoValidator ellenőrzi. Ellenőrzi, hogy a kérést küldő felhasználó tagja-e a projektnek, amihez a fájl tartozik. Ha nem tagja, 403-as hibakóddal tér vissza. Ha tagja a projektnek, akkor lekéri a fájl adatait. Sikeres lekérés esetén 200-as státuszkód mellett visszaküldi a fájl utolsó szerkesztési adatait.

- **authenticateUser:**

- Megszólított végpont: -
- Bejövő paraméter: -
- Visszatérési érték: -
- Leírás: Ellenőrzi, hogy a felhasználó be van-e lépve

Modellek

File

- Írható mezők:
 - id (integer, elsődleges kulcs)
 - name (string)
 - content (string)
 - projectId (integer, idegenkulcs)
 - lastEditedUser (string)
 - lastEditedTime (string)
 - createdAt (dateTime)
 - updatedAt (dateTime)
 - projects (belongsTo Projects modell)
- A beforeCreate dekorátorban létrehozott setContent metódus a fájl létrehozásakor a content mezőt üres stringként hozza létre.

Project

- Írható mezők:
 - id (integer, elsődleges kulcs)
 - name (string)
 - ownerId (integer, idegenkulcs)
 - createdAt (dateTime)
 - updatedAt (dateTime)
 - owner (belongsTo User modell)
 - files (hasMany File modell)
 - members (manyToMany User modell és project_members kapcsolótábla)
- Az afterCreate dekorátorban létrehozott createMemberTable metódus automatikusan létrehozza a projekt tagjainak tábláját, a tulajdonost hozzáadva az első tagként.
- A beforeDelete dekorátorban létrehozott deleteProject metódus törli a projekthez kapcsolódó összes fájlt és eltávolítja a tagokat a project_members táblából.

Project_member

- Kapcsolótábla, a Project és a User modellek között biztosítja a kapcsolatot
- Eltárolt mezők:
 - projectId (integer, idegenkulcs)
 - userId (integer, idegenkulcs)

User

- Az AuthFinder titkosítja a felhasználó jelszavát
- A felhasználói azonosítás e-mail címen alapul
- Írható mezők:
 - id (integer, elsődleges kulcs)
 - firstName (string)
 - lastName (string)
 - email (string)
 - password (string, hashed)
 - createdAt (dateTime)
 - updatedAt (dateTime)
 - projects (hasMany Project modell)
 - projectMembers (manyToMany Project modell és project_members kapcsolótábla)
- A beforeDelete dekorátorban megírt deleteUser metódus eltávolítja a felhasználót az összes projektből, ahol tag, lekérdezi a felhasználó által létrehozott projekteket és eltávolítja annak tagjait, majd törli magát a projektet is.

Tárgymutató

- **Array:**
 - Olyan lista, melynek elemei ugyanolyan adattípusból állnak.
- **Autentikáció:**
 - A felhasználók azonosításának és hitelesítésének folyamata, hogy ellenőrizzük, jogosultak-e egy adott erőforrás elérésére.
- **Backend:**
 - Egy alkalmazás azon része, amely a felhasználói felület mögött fut, és az adatbázishoz, fájlrendszerhez vagy más szolgáltatásokhoz kapcsolódik.
- **Bejövő paraméter:**
 - A függvénynek vagy metódusnak átadott adat.
- **BelongsTo:**
 - Kapcsolatot jelöl egy objektum és egy másik objektum között, ahol az előbbi az utóbbinak alárendelt.
- **Callback függvény:**
 - Olyan függvény, amelyet egy másik függvény hív meg a feldolgozás befejezésekor, lehetővé téve aszinkron programozást, moduláris kódot, újrafelhasználhatóságot és hibakezelést.
- **DateTime:**
 - Dátum és időpont tárolására használt adattípus.
- **Dekorátor:**
 - Olyan funkció, amely egy másik funkciót módosít.
- **DELETE kérés:**
 - Adat törlésére használt HTTP-kérés.
- **Elsődleges kulcs:**
 - Egy adatbázis táblában az egyedi azonosító mező.
- **Forráskód:**
 - Egy program általános formában írt utasításai.
- **Frontend:**
 - Egy weboldal felhasználói felülete.
- **GET kérés:**
 - Adat lekérésére használt HTTP-kérés.
- **Hashed:**
 - Egy algoritmus által átalakított adat, amelyet biztonságosabbá tesz a tároláshoz.

- **HasMany:**
 - Kapcsolatot jelöl egy objektum és egy másik objektum között, ahol az előbbi több utóbbihoz kapcsolódik.
- **Headers:**
 - HTTP-kérésben vagy válaszban további információkat tartalmazó adatok.
- **Hibakód:**
 - Egy hiba típusát jelző szám.
- **Hosting:**
 - Weboldal vagy alkalmazás üzemeltetése szerveren.
- **Hostnév:**
 - Egy számítógép neve a hálózaton.
- **Idegenkulcs:**
 - Egy adatbázis táblában egy másik táblára mutató mező.
- **Integer:**
 - Egy egész számot tároló adattípus
- **JSON:**
 - Adatok strukturált tárolására használt szöveges formátum.
- **Kapcsolótábla:**
 - Több adatbázis tábla közötti kapcsolatot tároló tábla.
- **Komponens:**
 - Egy felhasználói felület újrafelhasználható építőköve.
- **Kontroller:**
 - Az üzleti logikát megvalósító osztály, amely a bejövő kérésekre reagál.
- **LocalStorage:**
 - A böngészőben tárolt adatok, amelyek a böngésző bezárása után is megőrződnek.
- **ManyToMany:**
 - Kapcsolatot jelöl egy objektum és egy másik objektum között, ahol az előbbi több utóbbihoz, és az utóbbi több előbbihez kapcsolódik.
- **Metódus:**
 - Egy osztályon belüli funkció.
- **Modell:**
 - Adatokat reprezentáló osztály.
- **Objektum:**
 - Egy adott osztály példánya.
- **POST kérés:**
 - Adat beküldésére használt HTTP-kérés.

- **Service:**
 - Egy üzleti logikát megvalósító osztály, amelyet más osztályok használhatnak.
- **Státuszkód:**
 - Egy HTTP-válasz sikerét vagy hibáját jelző szám.
- **String:**
 - Szöveget tároló adattípus.
- **Token:**
 - Biztonsági azonosító, amely hitelesítéshez és jogosultság ellenőrzéshez használható.
- **URL:**
 - Egy weboldal vagy erőforrás címét meghatározó szöveges karakterlánc.
- **Validátor:**
 - Olyan objektum, amely ellenőrzi az adatok érvényességét.
- **Végpont:**
 - Az API által elérhető funkció vagy erőforrás.
- **Visszatérési érték:**
 - Egy függvény vagy metódus által visszaadott adat.

Tesztelés

A mobil applikáció tesztelésének tervezett dátuma: 2024. 04. 21.

Tesztelő: Sólyom Laura

Ábrajegyzék

1.1 ábra: Webes felület kezdetleges terve	4
1.2 ábra: Webes felület kezdetleges terve	4
2. ábra: Mobilos felület kezdetleges terve	5
3. ábra: Adatbázis terv	5

Felhasznált Irodalom

- Robert C. Martin (2021): Tiszta kód: Az agilis szoftverkézművesség kézikönyve
- CodeMirror 5 User Manual
 - <https://codemirror.net/5/doc/manual.html> (Utolsó letöltés: 2024. 04. 16.)
- Judge0 CE - API Docs
 - <https://ce.judge0.com/> (Utolsó letöltés: 2024. 04. 14.)
- Tailwind CSS - Documentation
 - <https://tailwindcss.com/docs/installation> (Utolsó letöltés: 2024. 04. 10.)
- Angular Documentation
 - <https://angular.io/docs> (Utolsó letöltés: 2024. 01.20.)
- AdonisJS Documentation
 - <https://v5-docs.adonisjs.com/guides/introduction> (Utolsó letöltés: 2024. 03. 20.)
- Ionic Documentation
 - <https://ionic.io/docs> (Utolsó letöltés: 2024. 02. 15.)

Köszönetnyilvánítás

Szeretnénk köszönetet mondani tanárainknak, Rékási Józsefnek (backend programozás), Bálint Dezsőnek (adatbázis-kezelés), Sallai Andrásnak (asztali és mobil alkalmazások fejlesztése), Jáger Attilának (frontend programozás) és Horváth Boglárkának (informatikai szakmai angol).

Köszönjük, hogy hozzájuk bármilyen kérdéssel és problémával fordulhattunk, segítségük nélkül nem lennénk most ott, ahol.