**Project\_MK**

**Előírások**

Tartalom

[Github 2](#_Toc100426188)

[**Token igénylése:** 2](#_Toc100426189)

[**Csatlakozás a Repository-hoz:** 2](#_Toc100426190)

[Adatbázis 4](#_Toc100426191)

[**Szerkezet:** 4](#_Toc100426192)

[**Leírás:** 4](#_Toc100426193)

[Frontend 5](#_Toc100426194)

[**Leírás:** 5](#_Toc100426195)

[**index.html:** 5](#_Toc100426196)

[**registration.html:** 6](#_Toc100426197)

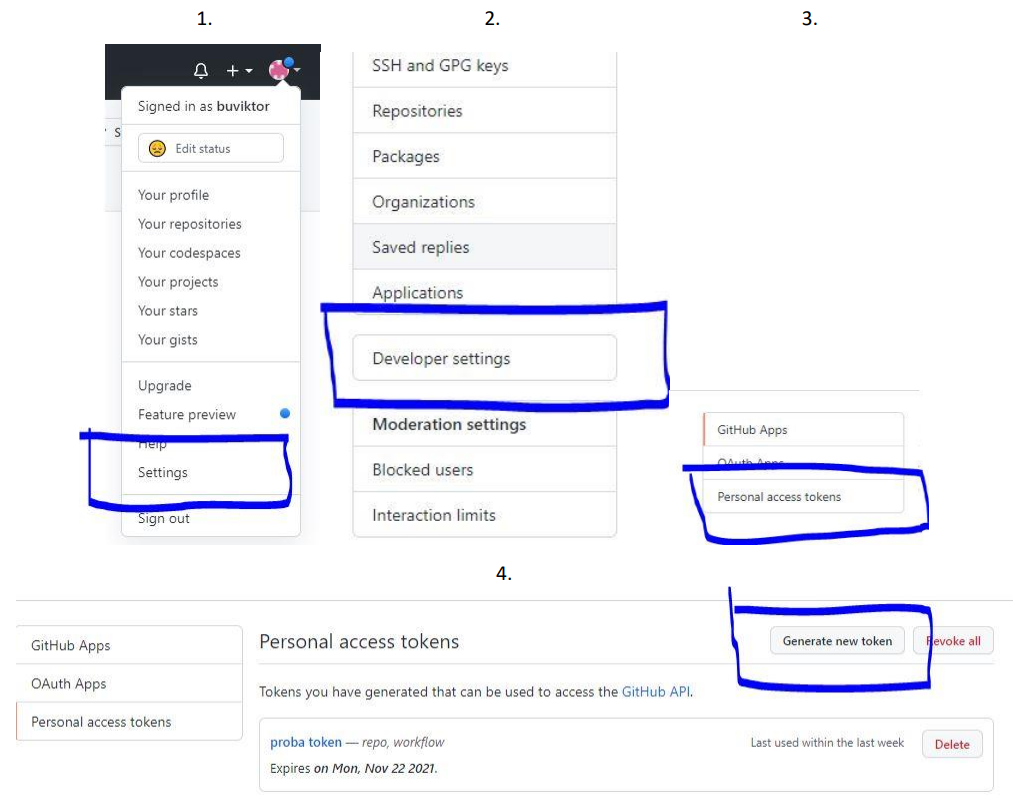
[**forgot.html:** 6](#_Toc100426198)

[**login.html:** 7](#_Toc100426199)

# Github

## **Token igénylése:**

Bejelentkezés után első lépésként igényelni kell token-t, hogy később lehessen csatlakozni a projekthez. Ezt a következő képen lehet megtenni:



Ezek után egy pipálós rész következik, én mindent bepipáltam a biztonság kedvéért.

**Majd a kész legenerált token kódját másold ki magadnak, valahova mert később szükség lesz rá a csatlakozáshoz!**

## **Csatlakozás a Repository-hoz:**

A repository a következő címen van:

<https://github.com/buviktor/Project_MK.git>

A csatlakozása a következő képen néz ki:

1. Ki kell választani a mappát ahová leakarod másolni, majd ott futtatni a GitBash-t (jobb klikk -> Git Bash Here). (Pl.: ha az asztalt választod, akkor majd ő fogja létrehozni a Repository nevével egy mappát és abba fogja beletenni a dolgokat. Ha esetleg nem az asztalra szeretnéd, akkor így néz ki a dolog D:/projektek/<ide jobb klikk>)
2. Majd ha megnyitottad a Gitbash-t a következő parancsok kellenek:
   * *git config --global user.email ”<regisztrált email cím>”*
   * *git config --global user.name ”<neved>”*
   * *git clone* [*https://github.com/buviktor/Project\_MK.git*](https://github.com/buviktor/Project_MK.git)
   * *cd Project\_MK/*
   * *git remote add Project* [*https://github.com/buviktor/Project\_MK.git*](https://github.com/buviktor/Project_MK.git)

Pontosan nem tudom, de valahol kérni fogja, hogy hitelesítsd magadat. Egy pici felugró böngésző ablak lesz. Itt kiválasztod, hogy token-nel szeretnél hitelesíteni és bemásolod a token-t amit létrehoztál. Elvileg, ha végig mentél a parancsokon, akkor már a legfrissebb változatát kell, hogy lásd. De érdemes minden munka megkezdése előtt frissíteni rajta ezt a következő paranccsal teheted meg:

* *git pull Project main*

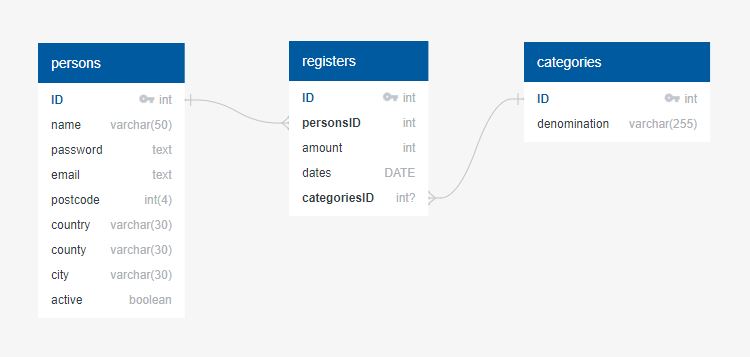
Ha valamit változtattál a következőkkel töltheted fel, hogy mindenki lássa:

* *git status* (ellenőrzöd, hogy valóban csak a módosított fájlok várnak feltöltésre)
* *git add <fájl>* (hozzáadod, a színpadhoz a feltölteni kíván fájlokat. Én úgy szoktam, hogy az add rész után csak egy tab-ot nyomok és oda teszi azt a mappát, amiben módosítva lett/lettek a fájl/fájlok. Ez is bőven elég neki, hogy hozzá adja a színpadhoz.)
* *git commit –m ”<kommented>*” (véglegesíted egy kommenttel, ha nem szeretnél kommentet írni, akkor csak egy pontot tegyél a két idézőjel közé)
* *git push –u Project main* (csak elsőnek utána elég lesz csak a *git push* parancs!)

Ha nem talántán ütközés van, azt jelezni fogja a fájlban! Kitörlöd a határoló jeleket és/vagy szöveget, majd a sorrenden alakítasz, amit ütközik.

# Adatbázis

## **Szerkezet:**



## **Leírás:**

**persons**  **Személyek tábla**

ID Egész szám, a felhasználók sorszáma, PK

name Szöveg, a felhasználó neve

password Szöveg, a felhasználó jelszava

email Szöveg, a felhasználó email címe

postcode Egész szám, a felhasználó irányítószáma

country Karakter, a felhasználó országa

county Karakter, a felhasználó megyéje

city Karakter, a felhasználó városa

active Logikai érték (0,1), a felhasználó fiókjának státusza

**regiszters** **Regisztrál adatok tábla**

ID Egész szám, a regisztrál adatok sorszáma, PK

personalsID Egész szám, a felhasználó sorszáma, FK

amount Egész szám, az adatok összeg formája (Ft-ban)

dates Dátum, az adatokhoz tartozó dátum

categoriesID Egész szám, a kategóriák sorszáma, FK

**categories** **Kategóriák tábla**

ID Egész szám, a kategóriák sorszáma, PK

denomination Szöveg, a kategóriák megnevezése

Az *elsődleges kulcsokat PK-val, az idegen kulcsokat FK-val van jelölve! A mező nevek után annak típusa és leírása található!*

Az adatbázishoz tartozó fájlokat a *Github Repository/Docs/Description* található.

Az amount mező előjeles egész számként tárolja az összegeket, annak függvényében kiadás vagy bevétel. Az active mező két értéket vehet fel. A ’*0’* a logikai hamis (false) és az ’*1*’ a logikai igaz (true).

A másodlagos kulcsok úgy kötöttem össze, hogy egy adott felhasználó törlésénél minden hozzá tartozó adatot töröljön a többi táblából is!

# Frontend

## **Leírás:**

Tartalmaznia kell egy regisztrációs és bejelentkezős részt, valamint egy pillanatképes Slide-ot és egy rövid leírást az oldalról.

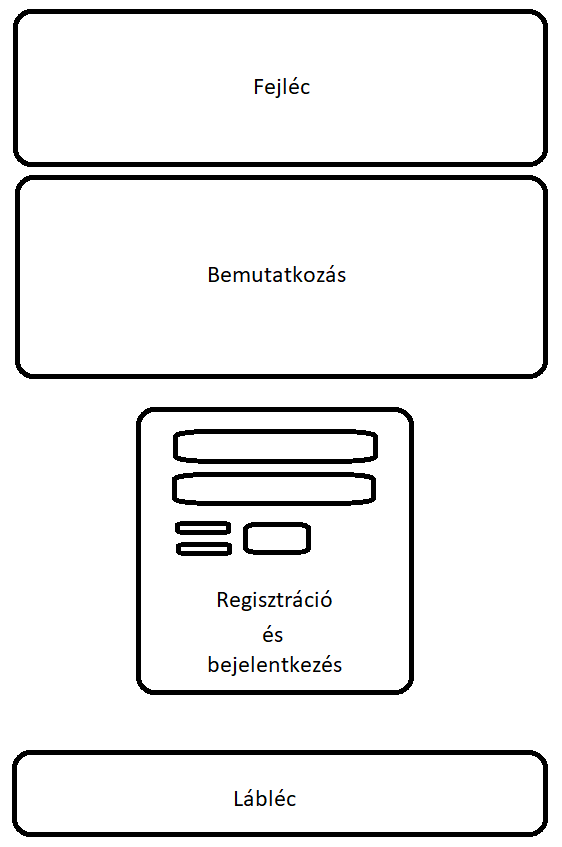
Szükséges felhasználni a következőket:

* HTML5
* Bootstrap5
* CSS (ha szükség van rá!)
* Javascript
* Reszponzív megjelenés

Szeretném, ha a lekérdezések mind listaszerkezetűek lennének, valamint magának az oldalnak lenne verzió kezelése. Ez alatt azt értem, hogy a Docs/Patch\_notes/ mappában lévő Changelog.txt fájl havi bontásban vezetve lenne. Ez két dolog miatt is előnyös:

1. Ha valaki elolvassa, tudja mi változott
2. komplexebb képet mutat a kivitelezés stádiumairól

## **index.HTML:**

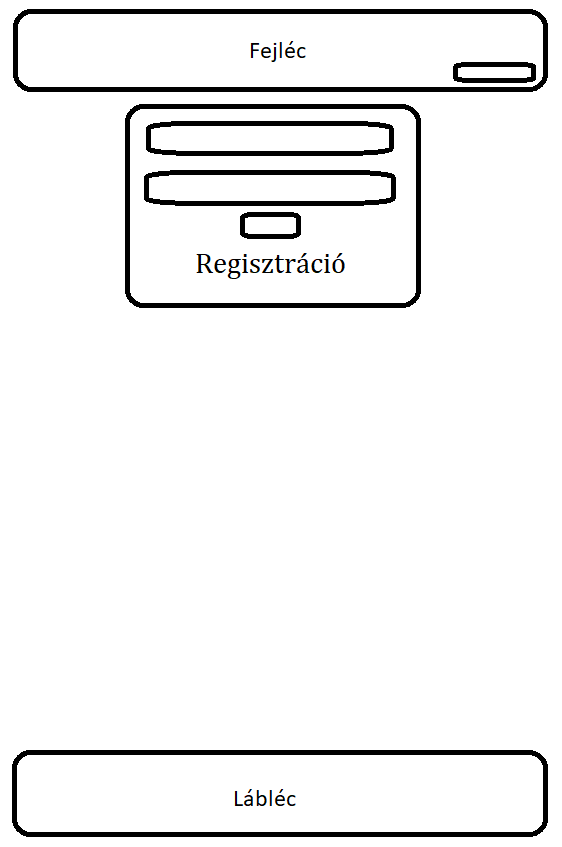
Leírás:

* Fejléc tartalmazza a Slide-ot a pillanatképekkel.
* Bemutatkozás rész lesz a rövid leírás az oldalról.
* Regisztráció és bejelentkezés részben a következők foglalnak helyett:
  + Felhasználónév
  + Jelszó
  + Elfelejtett jelszó link
  + Regisztráció link
  + Bejelentkezés gomb
* Lábléc részen pedig készítők neve és Github link (képként) lesz elhelyezve.

## **registration.html:**

Leírás:

* Fejléc tartalmazza a „Regisztráció” felíratott valamint egy linket, ami vissza visz az index.html-re.

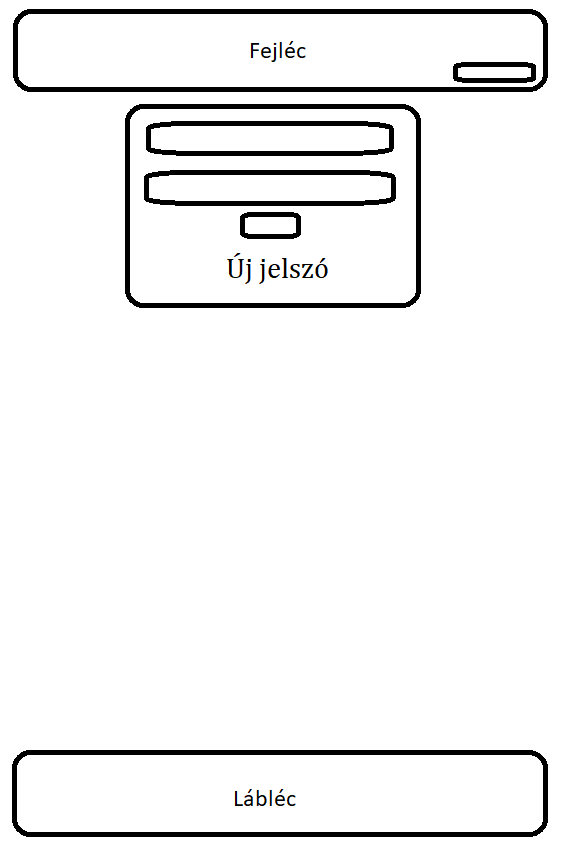


* Regisztráció részben a következők foglalnak helyett:
  + Felhasználónév
  + Jelszó
  + Regisztráció gomb
* Lábléc részen pedig készítők neve és Github link (képként) lesz elhelyezve.

## **forgot.html:**

Leírás:

* Fejléc tartalmazza a „Jelszó változtatás” felíratott valamint egy linket, ami vissza visz az index.html-re.

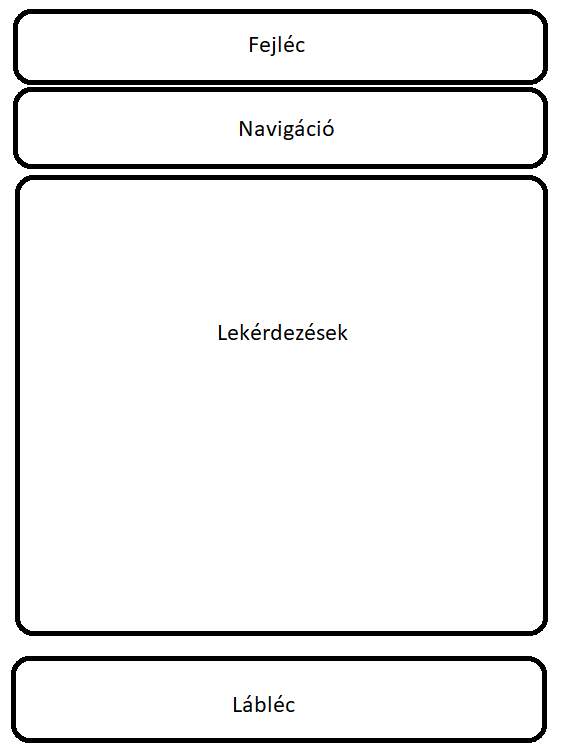


* Regisztráció részben a következők foglalnak helyett:
  + Felhasználónév
  + Jelszó
  + Új jelszó gomb
* Lábléc részen pedig készítők neve és Github link (képként) lesz elhelyezve.

## **login.html:**

Leírás:

* Fejléc tartalmazza az Üdvözlő szöveget a felhasználó nevével.



* Navigációs sávon helyezkednek el a következők:
  + Főoldal (login.html)
  + Részletes lekérdezés (query.html)
  + Hozzáadás (add.html)
* Lekérdezések részen havi bontásban megjelenik az összes adat amit felvitt. Itt a lekérdezések módosíthatóak lesznek, ill. törölhetőek.
* Lábléc részen pedig készítők neve és Github link (képként) lesz elhelyezve.

# Backend

## **Leírás:**

Jelszó kezelés az adatbázisban valósul meg. A jelszó min. 4 karakterből kell állnia és max. 10 karakter hosszú legyen.

Szükséges felhasználni a következőket:

* REST api
* JSON Token
* Node.Js (express, cors, mysql)

Szeretném, ha a lekérdezések mind listaszerkezetűek lennének, valamint magának az oldalnak lenne verzió kezelése. Ez alatt azt értem, hogy a Docs/Patch\_notes/ mappában lévő Changelog.txt fájl havi bontásban vezetve lenne. Ez két dolog miatt is előnyös:

1. Ha valaki elolvassa, tudja mi változott
2. komplexebb képet mutat a kivitelezés stádiumairól

## **Elérési utak:**

* /user/<userID> -- személyhez tartozó összes adat lekérdezése
* /user/<userID>/new/ -- személyhez tartozó új adat táblához hozzáadás
* /user/<userID>/<postID>/edit/ -- személyhez tartozó adat módosítása
* /user/<userID>/<postID>/delete/ -- személyhez tartozó adat törlése
* /user/<userID>/choose/date -- Lérdezés választás: dátum bontás
* /user/<userID>/choose/category -- Lérdezés választás: kategória bontás