Fejlesztési dokumentáció

EasyEffort

**Témavezető: Készítette:**

Szabó Ákos Bartha János

Inántsy-Pap Tamás Lukács Donát

Kecskés Krisztián Dávid

NYÍREGYHÁZA

2023

# 

Tartalomjegyzék

[**1.Projektterv**](#_1ubhh98d3i5l)

[**2.A projekt fontosabb lépései**](#_ikg8ukfkadxg)

[**3.Felmerült problémák és azok megoldásai**](#_cah4am4gg722)

[3.1 A probléma, amire a weboldalunk megoldást nyújt](#_7rgs5kr1ys35)

[3.2 Felmerült problémák a fejlesztés során](#_h5ouwanhkcnl)

# 

# 

# 

# 1.Projektterv

| **A feladat pontos leírása, amely tartalmazza a probléma teljes körű leírását, amelyre a vizsgaremekben készített alkalmazás megoldást ad.** | |
| --- | --- |
| Magyarországon az elhízottak csoportjába tartozik a teljes lakosság 23,9 százaléka, nemekre lebontva minden ötödik nő (19,9 százalék) és minden negyedik férfi (25,1 százalék). A *Easy Effort* segít azokon az embereken, akik fogyni szeretnének vagy csak egyszerűen egészségesebben élni.Külső személy validálásával/tanulmány alátámasztásával képesek leszünk meghatározni a felhasználók kalóriigényét, illetve azt nyomon követni.A felhasználó ki tudja választani majd az általa elfogyasztott élelmiszert, aminek az adatait egy külső API-n keresztül kapjuk majd,ami tartalmazza a táplálék makróit is.Lehetőség lesz még különböző edzéstervek megtekintésére(home,gym, streetworkout),amit még nemek szerint is bontunk.Az oldalon mindezekmellett lesz egy szekció a különböző táplálékkiegészítőkleírására esetleg ajánlására. | |
| **A feladattól elvárt követelményrendszer:** | |
| A felhasználónak lehetősége lesz regisztrálni.Miután ezt megtette, elérhetővé válik számára a kalória számláló, illetve az edzéstervek menüpont.A felhasználó regisztráció után bizonyos ideig nyomon tudja követni a napi kalóriabevitelét.A felhasználókat illetve a kalória bevitelt az adatbázis tárolja. | |
| **A feladat megoldása során használni kívánt architektúrák:** | |
| Adatbázis – MySQL  Backend – NodeJS  Frontend – ReactJS | |
| **A project időbeli ütemezése, a részekhez rendelt felelős megnevezésével** | |
| 2022.12.31. | Adatgyűjtés a projekthez,projektnapló létrehozása. Felelős:Lukács Donát  Design megtervezése. Felelős:Bartha János |
| 2023.01.31. | Adatbázis megtervezése és táblák létrehozása.Felelős:Kecskés Krisztián  Projektnapló és dokumentáció ellenőrzése,Backend – API feldolgozása.Felelős : Lukács Donát  Frontend – Alap váz létrehozása.Felelős : Bartha János |
| 2023.02.28. | Backend – Alapvető kérések feldolgozása,projektnapló ellenőzése. Felelős:Lukács Donát  Frontend – Az oldal megvalósítása a design szerint(70-80% -ra)  Felelős: Bartha János |
| 2023.03.31. | Backend - Befejezés, tökéletesítés, tesztelés Felelős: Lukács Donát  Frontend – Befejezés, tesztelés Felelős:Bartha János Egységek integrálása. Felelős: Kecskés Krisztián |
| 2023.04.15. | Az oldal végső tesztelése.  Dokumentációk ellenőrzése. Felelős:Kecskés Krisztián |

# 

# 2.A projekt fontosabb lépései

Az első lépés az adatgyűjtés volt, ami a terv szerint, december végén, január elején meg is történt, a teljes csapat részvételével.

A design tervek elkészítésével kicsit későn lettek elkészítve, de ez a fejlesztést nem befolyásolta.

A második fontosabb lépés az adatbázis megtervezése volt. Ezt a jelentkezési lapon feltűntetett időintervallumban teljesítette is Kecskés Krisztián. A backend API implementálása is megkezdődött terv szerint, amiért Lukács Donát volt a felelős, valamint Bartha János megkezdte a frontend alapköveinek a lepakolását is.

A következő, előre megfogalmazott mérföldkő február vége volt. A backend API kidolgozása ekkorra elérte a 70% körüli állapotot, amin továbbra is Lukács Donát dolgozott. A frontendel kicsit megcsúszott a csapat, ezért oda is besegített és elkezdte kidolgozni a kalória számláló oldalt. A tervek szerinti 70-80%-hoz képest a frontend visszamaradt, habár ez egy meredek döntés volt számunkra amikor ezt írtuk a jelentkezési lapon. Lukács Donát mellett, Bartha János és Kecskés Krisztián is a frontenden dolgoztak ebben az időszakban.

Március, a következő állomás. Terv szerint, a backend az nagyjából teljesen kész lett, pár apró módosítást kellett ezután már csak végezni rajta, hogy a frontendnek megfelelő formában küldje vissza az adatot. Ezért Lukács Donát volt a felelős. Frontend oldalon elkészültek a maradék oldalak Bartha János és Kecskés Krisztián kivitelezésével, valamint az admin oldal Lukács Donát által. Ekkor a legtöbb funkcionalitás már működött frontend oldalon.

Az utolsó hónapban, ugyanis a mérföldköveket hónapok szerint osztottuk fel a csapattal, csúsztunk a tervhez képest. A backenden áttértünk a Sequelize nevű ORM-re ezért a tesztelést 15.- ig , ami a tervben szerepelt, nem tudtuk megkezdeni, valamint volt egy kisebb csúszás is frontend oldalon is és az adatbázist is át kellett kicsit dolgozni. A backendért és a frontend egy részéért Lukács Donát, a frontend maradék részéért Bartha János, az adatbázisért Kecskés Krisztián volt a felelős.

# 3.Felmerült problémák és azok megoldásai

## 3.1 A probléma, amire a weboldalunk megoldást nyújt

Magyarországon az elhízottak csoportjába tartozik a teljes lakosság 23,9 százaléka, nemekre lebontva minden ötödik nő (19,9 százalék) és minden negyedik férfi (25,1 százalék). Az Easy Effort segít azokon az embereken, akik fogyni szeretnének vagy csak egyszerűen egészségesebben élni. Külső modul meghatározásával képesek leszünk meghatározni a felhasználók kalóriaigényét, illetve azt nyomon követni. A felhasználó ki tudja választani majd az általa elfogyasztott élelmiszert, aminek az adatait egy külső API-n keresztül kapjuk majd, ami tartalmazza a táplálék makróit is. Lehetőség lesz még különböző edzéstervek megtekintésére(home,gym, street workout). Az oldalon mindezekmellett lesz egy szekció a különböző táplálékkiegészítők leírására, ajánlására.

A kalória számláló működése alapján a felhasználó, ki tudja választani, hogy milyen mértékben akar fogyni vagy adott esetben hízni, majd megadja a rá jellemző adatokat és a backend kalkulál neki egy napi kalória értéket. Ezután ha valamit az illető elfogyasztott, csak rákeres, beírja az adagot, elküldi és máris a fenti csíkon megjelenik a változás, ami ki is írja, hogy az adott napon mennyi kalóriája maradt az illetőnek, illetve ha már nem maradt. A **cron** nevű npm modullal minden nap éjfélkor nullázódik a napi kalória érték az adatbázisban, de ezelőtt egy **history** nevű táblába kilistázza a szervezünk, hogy később ez visszakövethető legyen az egyén számára.

Ezek mellett az edzéstervek oldalunk segít a felhasználónak, illetve látogatóinknak formába lendülni, illetve ezek hatására nem csak egészségesebbek lesznek, de nő a kalóriaigényük is, ha nehezükre esne betartani a limitet.

## 

## 3.2 Felmerült problémák a fejlesztés során

A fejlesztési folyamat nem telt zökkenőmentesen. Mind a hármunknak ez volt az élete első nagy projektje, valamint az ezzel párhuzamosan történő iskolában történő oktatás sem könnyítette meg helyzetünket. Első nagy probléma backend oldalon történt, ugyanis kalória adatainkat külső API-ból kapjuk, ami elég bonyolult struktúrában küldi vissza az adatot, amit fel kellett dolgozni Lukács Donátnak, aki főként a backendért volt felelős, de besegített a frontend oldalon is. Ezt az akadályt azonban egy kis idő múlva orvosolta.

Talán ide sorolnám a designt is, de ez csak kisebb probléma volt, mivel Bartha Jánosnak, a csapat frontendesének, remek szeme van a designhoz.

Az autentikációt JWT-vel oldottuk meg, amit cookieban küldtünk el. Ezzel akadt egy kis problémája Lukács Donátnak, azonban a mentorokkal meg tudta ezt beszélni, és ők elmagyarázták annak használatát.

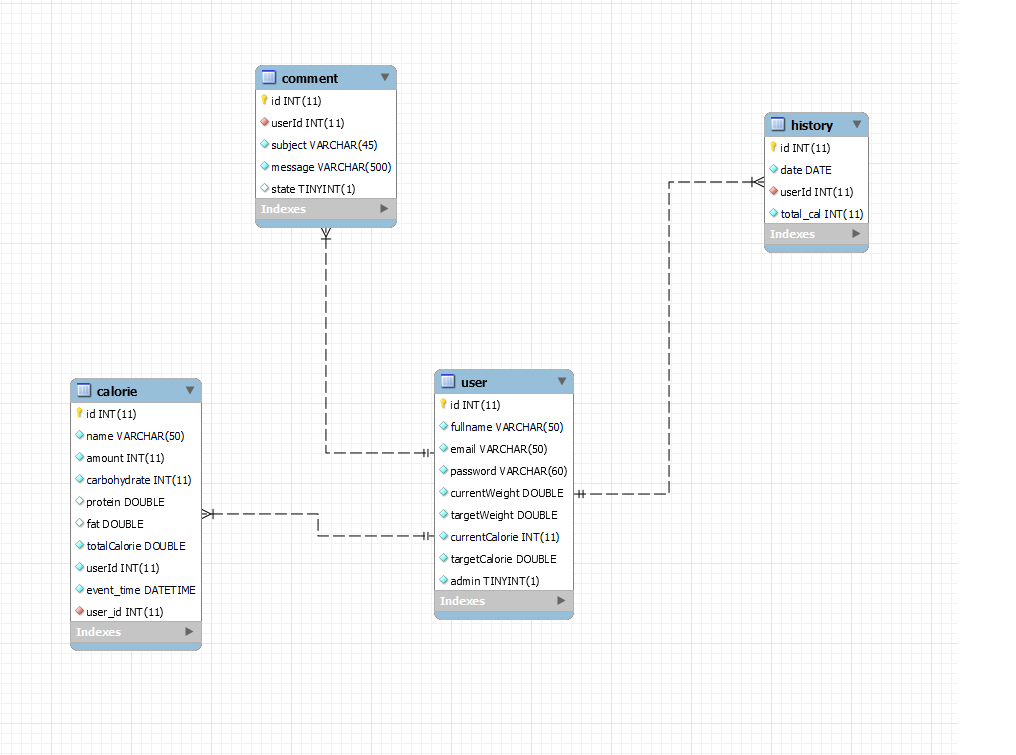
4.Fontosabb mentorálási események

A csapat mentorai, aktív tanáraink így nem csak órán kívül, de órákon is tudtunk útmutatást vagy segítséget kérni egy problémával kapcsolatban, vagy csak fejlesztési tanácsokat.

Mentoraink heti 2-3 szor ellenőrizték a projekt haladását, valamint ha problémánk akadt, azt is ekkor beszéltük meg velük.

A két csapat mentor rettentő segítőkészen állt fejlesztési munkánkhoz, segítettek amiben csak tudtak.

5.Adatbázismodell és a séma



Adatbázis modell séma

6. A projekt haladása, az egyes részfeladatoknál a tagok feladatának rövid leírását

6.1 A projekt haladása

A projekt, ahogy már a 2. pontban leírtam, a tervek szerint haladt. Kisebb csúszások voltak a határidőhöz képest, de ez betudható annak, hogy nagyon sok dolgot életünkben először csináltunk.

Áprilisban jött egy nagyobb csúszás, ahol kicsit elkalkuláltuk tervezésnél magunkat, és megcsúsztunk, de még határidőn belül maradtunk.

A backend rész, aminek kizárólagos felelőse Lukács Donát, február végére, elérte a 70% körüli szintet. Mivel a frontendel nem olyan ütemben haladott a csapat, amilyenben tervezve volt ezért a backend után oda is besegített, a csapat is belátta, hogy a frontend nagyobb részt emészt fel a projektből, mint arra számítottak.

Április közepére már csak a kisebb hibák javítása, illetve a tesztelés maradt hátra a projektből.

6.2 Egyes részfeladatoknál a tagok feladatának rövid leírása

* A backend teljes egészében Lukács Donát feladata volt, szóval ezt nem részletezném.
* Frontenden a Navigációs sávot Bartha jános készítette el, valamint ő dolgozta ki a színpalettát és a fő dizájnt.
* Szintén frontenden a főoldalt, a “contacts” oldalt, a footert és az “about us” oldalt Kecskés Krisztián és Bartha János közösen csinálták meg
* Az adatbázis tervét Lukács Donát és Kecskés Krisztián közösen beszélték meg, Kecskés Krisztián létre is hozta azt a terv alapján.
* A kalória számláló oldalt, bejelentkező és regisztrációs felületet Lukács Donát oldotta meg.
* A backend nullázása miatt minden éjfélkor, ahogy fentebb már említettem, az adatbázist kicsit módosítani kellett, aminek tervéért Lukács Donát, megvalósításáért Kecskés Krisztián volt a felelős.
* Backenden a Sequelize nevű ORM bevezetéséért, amivel sokkal hatékonyabbak és egyszerűek az adatbázis lekérdezések Lukács Donát volt a felelős.
* Backend tesztelés Postman segítségével. A backend route-ok tesztelésével kiszűrhetőek a felmerülő hibák. A teszt file megtekinthető a github repository-ban. A tesztet Lukács Dnát végezte el.

7.A felhasznált műszaki eszközöket, a felhasznált szoftvertechnológiákat, szabványokat és ajánlásokat.

7.1 Műszaki technológiák, szoftvertechnológiák

A backend megvalósítására a **NodeJS-**re esett a választásunk, azon belül is az **ExpressJS** nevű keretrendszerére. Ezzel a technológiával viszonylag gyorsan és egyszerűen lehet felállítani szerver oldali alkalmazást/API-t. Főként egyszerűbb, kevésbé bonyolult rendszerek kialakítására lehet jó választás. Mellette szólt még az az érv, hogy sokkal kezdő barátabb, mint a Java Spring Boot, ami a másik választási opciónk lett volna.

Adatbázis téren a **MySQL** adatbázisra esett a választásunk. A feladathoz könnyebben használható, mint egy NoSQL adatbázis, illetve a csapat összes tagja MySQL-ben a legtapasztaltabb.

A **Xampp** nevű programmal hostoltuk az adatbázisunkat lokálisan. Használatának oka az egyszerűsége, valamint, hogy tanulmányaink során főként ezt használta a csapat.

Frontend részen a **ReactJs** nevű JavaScript könyvtárra esett a választásunk. Utolsó éven kezdtük el tanulni, és iskola mellett sikerült azokat az alapokat elsajátítani, ami egy ilyen vizsgaremek elkészítéséhez kell. A React legfőbb jellemzője, hogy ún. “single page”, ami azt jelenti, hogy nem tölt újra az oldal.A Virtuális DOM-jának köszönhetően mindig csak azokat a komponenseket rendereli le, amik változnak az oldalon. Egy React app komponensekből áll, így támogatja a kód újrafelhasználását is.

A backend és frontend írásához is a **Visual Studio Code** nevű szoftvert használtuk. Rengeteg kiegészítője és letisztultsága miatt könnyen használható és képes volt futtatni egy időben a backendet és a frontendet is. A különböző bővítményei kényelmesebbé teszik a fejlesztési folyamatot.

A csapat megbeszélésekhez, közös kódoláshoz **discordot** használtunk, illetve ha csak üzeneteket küldtünk egymásnak akkor az facebookon, néha **emailben** történt.

A csapat a munka organizálásához, valamint verzió követéshez **GitHubot** használt. Szintén könnyíti a csapatban dolgozást, valamint ha valami hiba merült fel az új verziókban akkor könnyen vissza tudtuk azt fordítani egy régebbi verzióval. Könnyítette az iskolában vagy más helyről történő munkát is.

A **Trello** nevezetű weboldalt is használta a csapat. Itt általában a előre megtervezett ütemterveket, elkészült munkákat, amiket esetleg meg kellett beszélni mielőtt bekerültek töltöttük fel.

**NPM**-ről telepítettük a különböző olyan csomagokat, amik szükségesek a fejlesztéshez, vagy csak könnyítik azt, ugyanis React-al és Noda-al egyaránt használható.

7.2 Szabványok, ajánlások

A fejlesztés alatt törekedtünk betartani a **tiszta kód elvét.** Ez elősegítette a csapat munkát, ugyani a kód olvashatóbb lett és könnyebben karbantartható.

A fejlesztésünk a **RESTFUL architektúrára** épül. Az erőforrásoknak egyedi elérési útjai vannak, HTTP metódusokkal kommunikál a frontend a szerverrel és a metódus szabványokat betartjuk, az alkalmazás két részre bontható (kliens és szerver). Ennek az architektúrának egyik előnye, hogy egyszerű kommunikációt valósít meg és könnyen bővíthető.

Igyekeztünk betartani az **MVC** (Model View Controller) modellt. A Nézet részt a React, a Model részt a Sequelize nevezetű ORM, a Controller részért pedig az ExpressJs kontrollerei felelnek.

Adatbázis szinten törekedtünk a **3. Normálformára** hozásra, ezáltal könyebben kezelhetőek az adatok.

A teljes alkalmazás git repostiroy-ja: <https://github.com/lukacsdonat03/EasyEffort/tree/sequelize>