

DSZC MECHWART ANDRÁS GÉPIPARI ÉS INFORMATIKAI TECHNIKUM
•
SZOFTVERFEJLESZTŐ ÉS TESZTELŐ

FlexHUB

VIZSGAREMEK / PROJEKTMUNKA

KÉSZÍTETTE:

Fülöp Zsolt
Fazekas Gyula

Debrecen

2024

1. Bevezető

1.1. Téma ismertetése

Ez a projekt egy interaktív weboldal létrehozását tűzte ki célul, amely megkönnyíti a felhasználók számára az edzőtermek keresését és saját edzésterveik létrehozását. A felhasználók különböző szűrési lehetőségekkel böngészhetnek az adatbázisban található edzőtermek között, majd bejelentkezés után személyre szabott edzésterveket hozhatnak létre. A regisztráció és a bejelentkezés folyamata egyszerű és biztonságos, így a felhasználók könnyen hozzáférhetnek a weboldal nyújtotta összes funkcióhoz. A projekt célja egy átlátható és felhasználóbarát rendszer kialakítása, amely segíti a sportolni vágyókat céljaik elérésében.

1.2. Témaválasztás indoklása

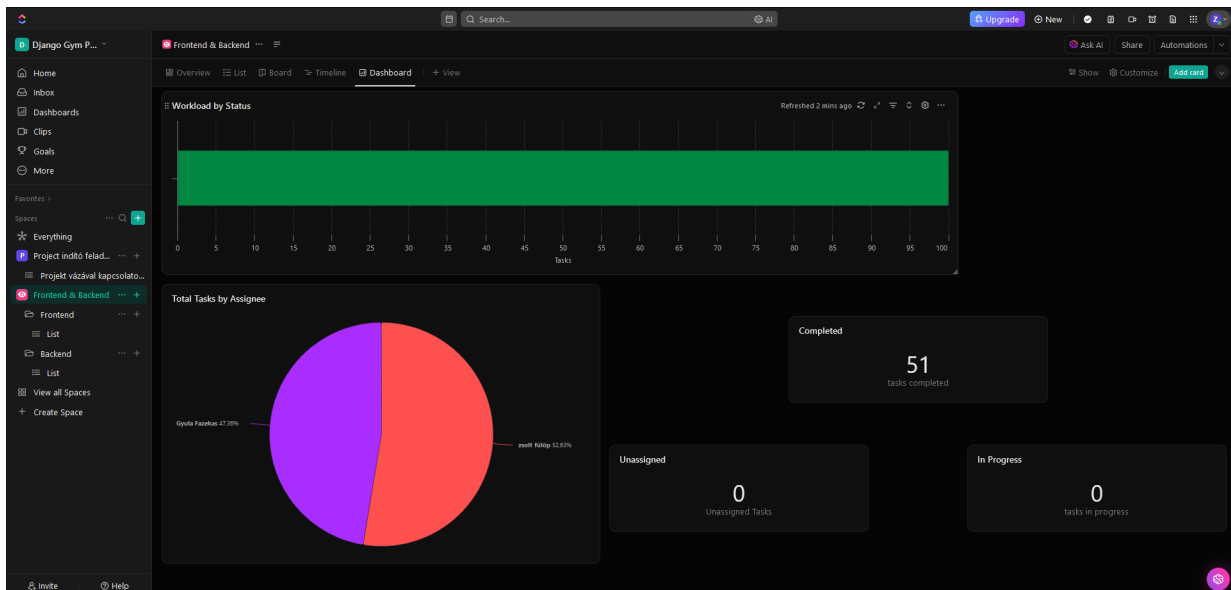
A projektünk témájának választásában jelentős szerepet játszott a személyes érdeklődésünk és tapasztalatunk az edzőtermek világában. Már évek óta rendszeresen edzünk és szenvedélyesen szeretjük ezt az életmódot, így adott volt számunkra, hogy olyan megoldást hozzunk létre, amely másoknak, különösen a kezdőknek segít eligazodni ebben a világban. Az edzőtermek keresésének és az egyéni edzéstervek kialakításának megkönnyítése fontos célkitűzésünk, mivel saját tapasztalataink alapján tudjuk, milyen kihívásokkal néznek szembe a kezdők. Projektünkkel egy átlátható és használható eszközt kívánunk biztosítani, amely támogatja a felhasználók edzési céljainak elérését és elősegíti az egészséges életmód kialakítását.

1.3. Felhasználói dokumentáció

Első napunk fő célja a szükséges programok összehangolása volt, hogy biztosítsuk a problémamentes és gördülékeny munkavégzést, akár külön-külön is dolgozhassunk. Alapvetően a **Visual Studio Code** szolgált a programunk alapjaként. A párhuzamos munkavégzés megkönnyítése érdekében a **GitHub**-ot használtuk, ahol létrehoztunk egy repozitóriumot (*repository*) három különálló ággal (*branch*): *main*, *fzs* és *fgy*. Az utóbbi kettőt a párhuzamos munkavégzés érdekében hoztuk létre. A szeparált ágak használata miatt gyakran használtuk az úgynevezett egyesítés (*merge*) GitHub parancsot, amely lehetővé tette számunkra, hogy szinkronizáljuk saját munkánkat a projekt fő ágával (*main branch*).



A kitűzött feladatok nyomon követése érdekében a **ClickUp** netes felületet használtuk, amely lehetőséget adott arra, hogy több különálló teret (*space*) hozzunk létre. Ezeken belül mappákat hoztunk létre, melyeket a projekt nagyobb részei szerint osztottunk szét, így két fő mappánk lett: *Projektindító feladatok* és *Frontend & Backend*. A mappákon belül táblákat hoztunk létre a kapcsolódó feladatokkal, így követhettük nyomon, hogy ki mikor készült el az egyes feladatokkal.



2. Projekt bemutatása

2.1. Előkészítés, tervezés

Miután tudtuk, hogy működnek a programok, amiket használni fogunk, elkezdtuk tervezni a projekteknek a felépítését. Több szempontot is figyelembe kellett vennünk a tervezési fázisban. Először is, hogy kik lesznek a célközönségünk, milyen problémával állhatnak szemben és azt hogyan lehet orvosolni, vagy egyszerűbbé tenni azt. Amint erre sikerült egy fajta megoldást kitalálni, tovább léphettünk a következő nagyobb feladatunkhoz, ami a funkcionalitás felépítése volt.

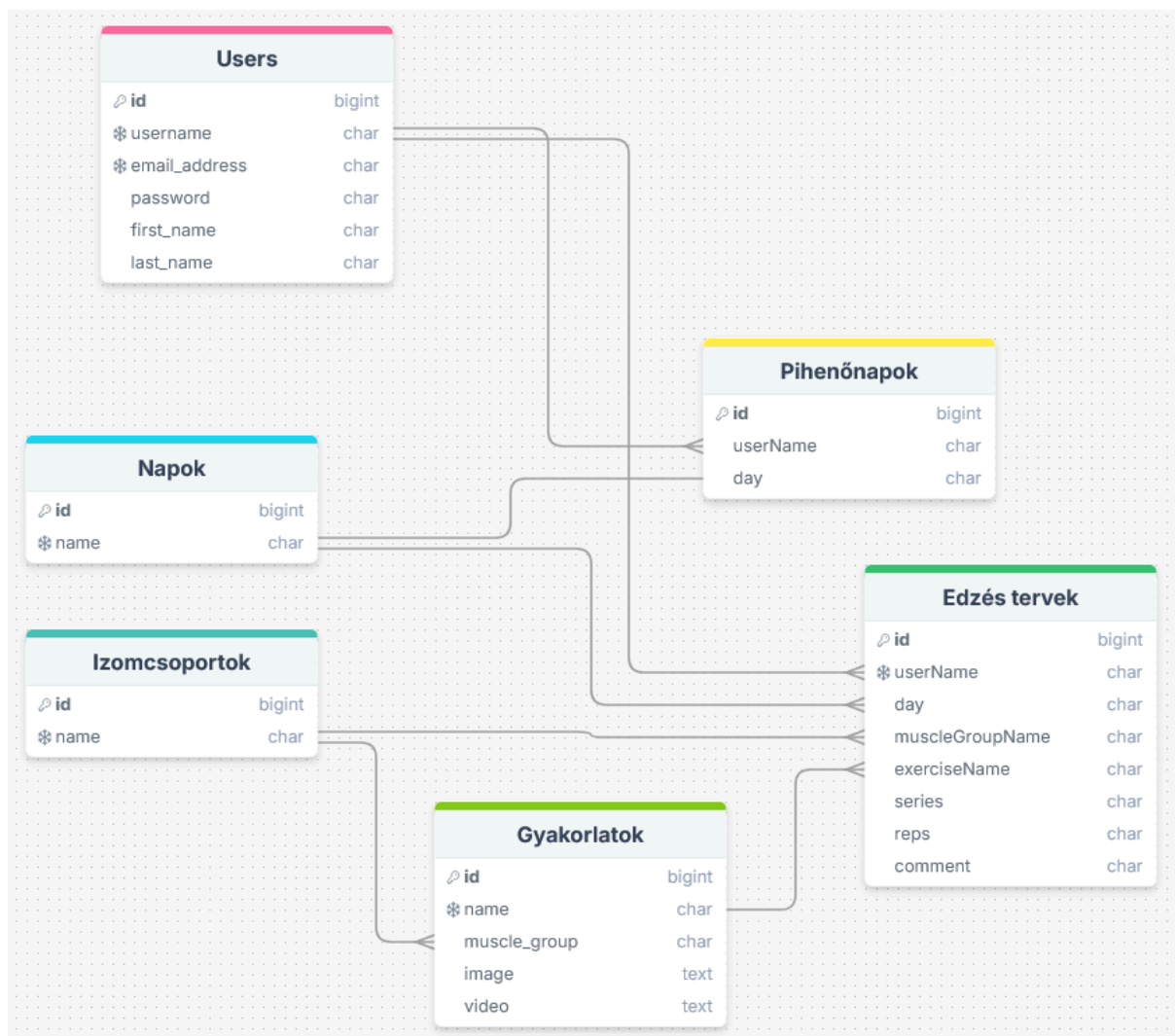
A vizsgaremekünk több oldalból épül fel, de az összes oldalunknak ugyanaz az alapja, amit (*base.html* és *base.css*) néven hoztuk létre. Továbbá létrehoztunk négy alap oldalt, illetve két olyan oldalt, ami csak 1-1 html elemet tartalmaz szeparálva a könnyebb integrálhatóság végett (*aboutUs.html*, *budgetBiceps.html*, *premiumPump.html*, *creatingWorkoutPlan.html*, *navbar.html*, *footer.html*).

Ezekén az oldalakon kívül, használtuk a Django beépített funkcióját a felhasználók regisztrálására, illetve bejelentkeztetésére, melynek sablon (*template*) fájljait a *registration* nevű mappába hoztuk létre (*register.html*, *login.html*, *password_reset_form.html*, *password_reset_complete.html*, *password_reset_confirm.html*, *password_reset_email.html*, *password_reset_done.html*).

Ahhoz, hogy tovább tudjunk haladni, szükségünk volt az adatbázisunk feltöltésére, ezért adatokat kellett gyűjtenünk, illetve létre kellett hoznunk az adatbázis rendszerünket. Először is az edzőtermek adatainak kezelésére fókuszáltunk, hogy megjeleníthessük őket az *Edzőtermek listázó* oldalunkon. Az adatokat az (*Edzőtermek*, *Edzőtermek címe*, *Városok*) adattáblákban tároltuk el.



Miután feltöltöttük a városok tábláját, összekapcsoltuk az edzőtermek címe táblával a városneveket felhasználva. Az edzőtermek címeit ezután az edzőtermek táblájával kötöttük össze. A regisztrációs és bejelentkezési oldalak adatainak mentéséhez a Django beépített funkcióit használtuk, így nem kellett új adattáblát létrehoznunk, minden új felhasználói adatot a beépített (*Users*) adattáblába mentettünk. Végezetül létrehoztuk az edzésstervekkel kapcsolatos adattáblákat, amit öt különálló adattáblával oldottunk meg (*Edzésstervek*, *Izomcsoportok*, *Gyakorlatok*, *Napok*, *Pihenőnapok*). Ezek közül az izomcsoportok és a napok adattáblái függetlenek. A gyakorlatokat az izomcsoportokhoz kapcsoltuk, a pihenőnapokat a napokhoz, az edzéssterveket pedig mindegyik adattáblával összekapcsoltuk a pihenőnapok kivételével. Emellett a gyakorlatok és a pihenőnapok adattáblái tartalmazzák a felhasználói neveket is. Ezek után, hogy feltöltöttünk minden fontos adattáblát, elkezdhetünk dolgozni az oldalaknak a funkcionalitásain, amiket javascript-el és python-al kellett megoldanunk.



2.2. Nehézségek

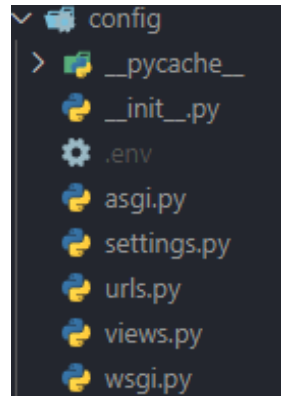
Első komoly problémánkkal gyorsan szembesültünk, és ez a GitHub használatához kapcsolódott. Összegyűjtöttük a nagyobb városokban található összes edzőtermet, amelyeket szerettünk volna az adatbázisunkban tárolni. Ezeket a változtatásokat a privát branch-ről a main branch-re kellett volna merge-ölnünk, de az egyesítés során konfliktusok (*merge conflicts*) léptek fel, amelyeket nem tudtunk megfelelően kezelni. Ennek eredményeként teljes adatvesztést szenvedtünk el, és újra kellett kezdenünk az edzőtermek adattáblájának feltöltését.

Az első nehézség leküzdését követően hamarosan egy újabb problémával találtuk szembe magunkat. Azon az oldalon, ahol az edzőtermeket listázzuk és szűrési lehetőséget biztosítunk a felhasználóknak, a szűrési funkciók nem működtek megfelelően. Ezt a problémát csak a projekt kódolásának legvégén sikerült orvosolnunk.

Továbbá, nehézségekbe ütköztünk a beépített Django regisztrációs funkciók használatával is. Először a stílus kialakítása jelentett kihívást, mivel ez egy beépített funkció, így közvetlenül nem tudtuk módosítani a megjelenését. Böngészőn belüli vizsgálatokkal kellett rájönnünk, hogyan épül fel a funkció. Azonban a Bootstrap osztályok használata nehezen vagy egyáltalán nem volt lehetséges, és saját osztályok vagy más jelölők használatát sem támogatta. Hosszas kutatások és az AI segítségével végül sikerült úgy megoldanunk ezeket a problémákat, hogy saját jelölők és Bootstrap osztályok használatát is lehetővé tegyük. Például a bejelentkező oldalon Bootstrap class használatával adtunk attribútumokat az input mezőknek, ahogyan az a képen is látható.

```
class LoginForm(AuthenticationForm):
    class Meta:
        model = User
        fields = ['username', 'password']
        widgets = {
            'username': forms.TextInput(attrs={'class': 'form-control'}),
            'password': forms.PasswordInput(attrs={'class': 'form-control'}) }
```

A Django beépített regisztrációs funkcióinak használata során további problémák merültek fel az elfelejtett jelszavakhoz tartozó e-mail küldésével kapcsolatban. Hosszas, sikertelen utánajárás és tesztelés után rájöttünk, hogy érdemes létrehozni egy `.env` nevű fájlt, amely az e-mail cím és az ahhoz tartozó applikáció jelszó tárolására vagy bármilyen érzékeny adat védelmére szolgál. Ezt a fájlt a `settings.py` fájl mellé kell elhelyezni, ahogyan az a képen is látható.



Továbbá a `(.env)` használatához szükséges egy `(dotenv)` nevű python könyvtárra is, amit a `settings.py`-ba kell importálni és ott betölteni a `(load_dotenv())` nevű parancsot használva. Illetve, ahhoz hogy az email küldés működjön további kóddal kell kiegészíteni a `settings.py` fájlt, ami a következőképpen néz ki.

```
mail = os.environ.get('MAIL')
mail_pw = os.environ.get('MAIL_PASSWORD')

EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = mail
EMAIL_HOST_PASSWORD = mail_pw
DEFAULT_FROM_EMAIL = mail
EMAIL_SUBJECT_PREFIX = '[FlexHub] Elfelejtett jelszó'
```

Az utolsó nagy problémánk, akkor jött elő, amikor már az edzés terveket menteni akartuk volna. Több ugyanolyan gyakorlatot ugyanahhoz a naphoz nem lehetett hozzáadni és a gyakorlatok sorrendjén se lehetett változtatni. Ahhoz viszont, hogy a felhasználó a végleges edzéstervében is lássa a gyakorlatok sorrendjének változását, akkor az adatbázisban is frissíteni kell a gyakorlatokat. A mi megoldásunk egy javascriptből indított (*fetch api*) használatával indul, amit (*post*) metódus segítségével (*json*) formátumban küldünk a backend részére, ahol python-ban beolvassuk ezeket az adatokat és regisztráljuk az adatbázisba őket. A gyakorlatok duplikálását és a sorrend módosításokat érintő hibákat úgy oldottuk meg, hogy először töröljük az aktuális felhasználóhoz és az aktuális naphoz tartozó összes már korábban elmentett gyakorlatot, majd újra – de most már a legfrissebb – gyakorlatok kerülnek mentésre.

```
document.querySelector("#save-workout-plan").addEventListener("click", () => {
  const workoutPlanData = GetFetchableWorkoutExerciseData();
  const csrftoken = document.querySelector('[name=csrfmiddlewaretoken]').value;
  const request = new Request(
    "http://127.0.0.1:8000/createWorkoutPlan/",
    {headers: {'X-CSRFToken': csrftoken}}
  );
  fetch(request, {
    method: "POST",
    body: JSON.stringify(workoutPlanData)
  }).then(() => {
    document.querySelector("#workout-plan-form").reset();
    window.location.reload();
  });
});
```

```
if request.method == "POST":
    exercisesJson = json.loads(request.body.decode('utf-8'))
    previousWorkoutPlan = workoutPlans.filter(
        userName=User.objects.get(username=exercisesJson[0]["user_name"]),
        day=days.get(name=exercisesJson[0]["day"]))
    newWorkoutPlan = []
    for exercise in exercisesJson:
        userName = User.objects.get(username=exercise["user_name"])
        day = days.get(name=exercise["day"])
        muscleGroup = muscleGroups.get(name=exercise["muscle_group_name"])
        exerciseName = exercises.get(name=exercise["exercise_name"])
        series = exercise["series"]
        reps = exercise["reps"]
        comment = exercise["comment"]
        if exerciseName not in newWorkoutPlan:
            newWorkoutPlan.append(WorkoutPlans(
                userName=userName, day=day, muscleGroupName=muscleGroup,
                exerciseName=exerciseName, series=series, reps=reps, comment=comment))

    previousWorkoutPlan.delete()

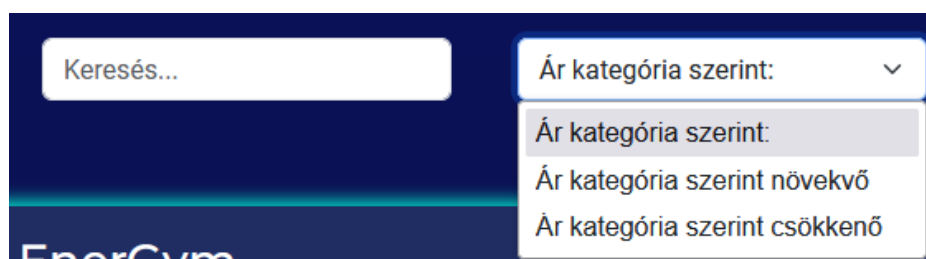
    for workoutPlan in newWorkoutPlan:
        workoutPlan.save()

    return redirect("PremiumPump")
```


2.3. A projekt funkcionalitása

Az oldalunkra látogatva a felhasználó először a fő oldalunkat fogja látni, ahol több navigációs menü közül is választhat, hogy hova szeretne tovább ugrani a weboldalon. A navigációs sáv bal szélén elhelyezkedik a logónk, majd mellette balról jobbra: *Főoldal*, *Edzőterem kereső*, *Rólunk*, *Bejelentkezés*.

Az edzőterem kereső oldalon a felhasználó először láthatja az összes olyan edzőtermet, ami az adatbázisunkban benne van, illetve egy kereső mezőt és egy legördülő menüt. Ez az előbb említett mező és menü segít a felhasználónak, hogy keresni tudjon az edzőtermek között. A felhasználó tud edzőterem névre, városra vagy olcsóbb és drágább edzőtermekre keresni. Ez utóbbit sávozással oldottuk meg, összesen négy ársávba soroltuk az edzőtermeket.



A következő oldal, aminek van funkcionalitása az a bejelentkező oldal, ahol az oldalunkon a korábban már regisztrált felhasználó a felhasználónevével és jelszával be tud lépni az oldalunkra. Amennyiben még nem regisztrált a felhasználó, akkor egy link segítségével erről az oldalról tovább tud menni a regisztrációs oldalra, ahol e-mail cím, felhasználónév, jelszó és a teljes nevének megadásával tud regisztrálni. Illetve a bejelentkező oldalon van egy további link, amivel az elfelejtett jelszava helyett új jelszót tud megadni az e-mail-ben kapott linken keresztül.

Végezetül pedig, ha a felhasználó sikeresen bejelentkezik, akkor egy új menüpont válik elérhetővé a számára, ami az edzésterv készítő oldal. Először is, amikor a felhasználó ezen az oldalon továbblép, hét üres táblázattal fog szembesülni, hiszen minden napra van lehetőség edzésterv hozzáadására. Minden napnál látható egy gomb (a képen jobb oldalt), amivel tovább tud menni a felhasználó arra az oldalra, ahol a gyakorlatokat hozzá tudja adni az edzéstervéhez. Illetve, ha aznap pihenő napot szeretne, akkor rákattint a gombra (a képen bal oldalt), és az aznapi táblázat el is tűnik.

Szerda

☐ Pihenőnap

Gyakorlatok módosítása/hozzáadása

Gyakorlat neve	Szettek	Ismétlések	Megjegyzés	Kép
----------------	---------	------------	------------	-----

A gyakorlatokat módosító és hozzáadó oldalon pedig a felhasználó meg tudja adni, hogy az egyes gyakorlatokból hány szettet és ismétlést szeretne csinálni. Még akár kis megjegyzéseket is adhat magának, amiket a későbbiekben is látni fog.

Vissza az edzés tervekre

Szerda

Izom csoport:
Válassz....

Szettek:

Ismétlések:

Megjegyzés:

Hozzáadás

Amint a felhasználó kiválasztotta, hogy melyik gyakorlatot szeretné hozzáadni, a hozzáadás gomb alatt meg fog jelenni egy kép a gyakorlatról. Ez azt a célt szolgálja, hogy ha a felhasználó nem ismeri fel az adott gyakorlatot a nevéből, akkor így be tudja azonosítani. A megjelenő képre kattintva tovább ugorhat egy új oldalra, ahol egy videót nézhet meg az aktuális gyakorlat szabályos végzésével kapcsolatban.

Izom csoport:

Hát

Gyakorlat:

Chin Ups (Fordított kezes felhúzás)

Szettek:

Ismétlések:

Megjegyzés:

Hozzáadás

Kattints a képre, ha érdekel a videó a gyakorlat bemutatásáról!



Amint kitöltötte a felhasználó az összes mezőt és rányomott a hozzáadás gombra, akkor az oldal jobb oldalán megjelenik a gyakorlat. Fontos tudni, hogy ilyenkor még nem menti el adatbázisba a gyakorlatot, csak megjeleníti jobb oldalt egymás alatt az összeset, amit hozzáad.

Itt drag-and-drop szerűen lehet változtatni a gyakorlatok sorrendjét, illetve törölni is lehet ha mégsem akarja a felhasználó hozzáadni az edzéstervéhez. A módosítás mentése gombra kattintva elmenti az adatbázisba és innentől kezdve az előző oldalon a táblázatban láthatóvá válik az összes hozzáadott gyakorlat.

1. Dumbbell Row Unilateral (Egyoldalú súlyzós evezés)

- Hát
- 10 szett
- 4 ismétlés
- Valami megjegyzés.

Törlés

2. Pull Ups (Felhúzás)

- Hát
- 10 szett
- 4 ismétlés
- Valami megjegyzés.

Törlés

3. Chin Ups (Fordított kezes felhúzás)

- Hát
- 10 szett
- 4 ismétlés
- Valami megjegyzés.

Törlés

Módosítás mentése

Illetve még egy funkció láthatóvá válik: a táblázat jobb szélső oszlopában a kis kép ikonra ha ráviszi az egeret a felhasználó, akkor megjelenik az adott gyakorlat képe kicsiben.

Szerda				
<input type="checkbox"/> Pihenőnap		Gyakorlatok módosítása/hozzáadása		
Gyakorlat neve	Szettek	Ismétlések	Megjegyzés	Kép
Dumbbell Row Unilateral (Egyoldalú súlyzós evezés)	10	4	Valami megjegyzés.	
Pull Ups (Felhúzás)	10	4	Valami megjegyzés.	
Chin Ups (Fordított kezes felhúzás)	10	4	Valami megjegyzés.	

3. Összegzés

Ez a projekt egy interaktív weboldal létrehozásáról szólt, amely meg tudja könnyíteni a felhasználók számára az edzőtermek keresését és saját edzésterveik megalkotását. A felhasználók különböző szűrési lehetőségekkel tudnak böngészni az adatbázisban található edzőtermek között, majd bejelentkezés után személyre szabott edzésterveket készíthetnek. A regisztráció és a bejelentkezés folyamata egyszerű és biztonságos, így a felhasználók könnyedén hozzáférhetnek a weboldal összes funkciójához. A projekt célja egy átlátható és felhasználóbarát rendszer létrehozása volt, amely támogatja a sportolni vágyók céljainak elérését.

Az egyik legnagyobb sikerünk, hogy elképzeléseink valósággá váltak. Egy közös projekten dolgozni csapattagként kihívásokkal és előnyökkel jár. Közösén felismertük, milyen egyszerű eszközökkel lehet megkönnyíteni mindennapjainkat, melyekre gyakran nem is gondolnánk. Számos kihívást tapasztaltunk, beleértve az egyidejű programfejlesztés megoldását mindkettőnk számára. Célunk egy átfogó, egyszerűen használható, mégis összetett rendszer létrehozása volt, amely minden felhasználó igényeit kielégíti, legyen szó hobbistákról vagy szenvedélyes testépítőkről. A weboldal lehetővé teszi a felhasználók számára, hogy időt spóroljanak, amikor a pénztárcájuknak megfelelő edzőtermet keresik lakóhelyük közelében, vagy hatékony gyakorlatokat keresnek különböző izomcsoportok edzéséhez.

Célunk az volt, hogy átlátható képet nyújtsunk az edzőknek és tanítványaiknak az edzéstervekről, a gyakorlatok végrehajtásáról és az érintett izmokról. Az alapokat úgy alakítottuk ki, hogy minden felhasználó számára elérhető legyen a megfelelő információ, beleértve az adatbázist, amely listázza a városok edzőtermeit és árait. Létrehoztuk a weboldal magját képező edzésterv funkciót, amely izomcsoportokra kereshető gyakorlatokkal van feltöltve. A gyakorlatok mind magyarul, mind angolul meg vannak nevezve, képekkel és videókkal a könnyebb kivitelezhetőség érdekében. A jövőben továbbfejleszthető a projekt, beleértve az étrendtervezést, vitaminok és táplálékkiegészítők használatát, valamint egy nagyobb közösség kialakítását.