

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Webová aplikace pro zaznamenávání tréninků a výkonů

Aleš Lahodný

Appollog

Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování

Třída: IT4

Školní rok: 2021/2022

Poděkování

Rád bych poděkoval pánům učitelům Ing. Petru Grussmannovi a Mgr. Marku Lučnému za jejich pomoc s projektem, jelikož mi poskytovali cenné rady a připomínky.

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2021

podpis autora práce

ANOTACE

Výsledkem projektu je funkční webová aplikace pro zaznamenávání tréninků a výkonů. Aplikace zahrnuje registraci uživatelů přes Google a Facebook, včetně standardních operací s uživatelským účtem jako je ověření či změna emailu nebo reset hesla. Uživatel si zaznamenává jednotlivá cvičení, a to s parametry jako jsou jméno, popis, pocit, míra námahy a hlavně datum. Stěžejní částí aplikace je totiž kalendář zobrazující jednotlivá cvičení. Umožňuje jejich filtraci podle druhu cvičení i textové vyhledávání. Při jeho návrhu bylo hlavní intuitivní ovládání a přehlednost. Důležitým parametrem cvičení je také jeho druh (sport, disciplína). Uživatel má po registraci k dispozici několik základních druhů, avšak může si přidat i své vlastní. Druhá část aplikace slouží k zaznamenávání výkonů. Uživatel si opět vytvoří disciplínu, tentokrát již více specifickou, a k ní si zaznamenává své výkony. Tyto data se mu následně zobrazují srozumitelnou vizuální formou v grafu. Aplikace má zcela responzivní design a funguje tedy na všech zařízeních.

Klíčová slova: webová aplikace, databáze, responzivní design, uživatelské účty, grafy

OBSAH

ÚVOD.....	5
1 TEORETICKÁ A METODICKÁ VÝCHODISKA	6
1.1 ELEKTRONICKÉ DENÍKY A POZNÁMKY	6
1.2 ARCHITEKTURA DATABÁZOVÝCH APLIKACÍ	6
1.3 POČÁTEČNÍ ZKUŠENOSTI	7
2 VYUŽITÉ TECHNOLOGIE.....	8
2.1 DJANGO CMS	8
2.2 POSTGRESQL	8
2.3 BOOTSTRAP 5	8
2.4 KNIHOVNA JQUERY	9
2.5 CHART.JS	9
2.6 DOCKER	9
3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY	10
3.1 ZALOŽENÍ PROJEKTU	10
3.2 DATABÁZOVÝ MODEL	10
3.3 ADRESÁŘOVÁ STRUKTURA.....	11
3.4 AUTENTIZACE A AUTORIZACE.....	11
3.4.1 BALÍČEK DJANGO-ALLAUTH.....	12
3.4.2 AUTENTIZACE POMOCÍ APLIKACÍ TŘETÍCH STRAN	12
3.4.3 ZÁKLADNÍ OPERACE S UŽIVATELSKÝM ÚČTEM	13
3.5 ZÁZNAMY TRÉNINKŮ	14
3.5.1 KALENDÁŘ.....	14
3.5.2 FILTROVÁNÍ	16
3.5.3 VYHLEDÁVÁNÍ.....	16
3.6 ZÁZNAMY VÝKONŮ.....	16
3.7 ÚVODNÍ STRANA	18
4 VÝSLEDKY ŘEŠENÍ, VÝSTUPY, UŽIVATELSKÝ MANUÁL	19
4.1 FUNKCE APLIKACE	19
4.2 SPLNĚNÉ A NESPLNĚNÉ CÍLE	20
ZÁVĚR	21
SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ	22

ÚVOD

Každý profesionální sportovec si vede řádný záznam o svých trénincích. Mít přehled ve trénincích zajišťuje jejich lepší správu, a tedy i lepší výsledky. V profesionálním sportu všichni trénují stejně tvrdě, hlavní rozdíly dělá trénink. Trenérství je skutečně věda. Rozvržení tréninků totiž musí zajišťovat udržitelnost (aby sportovec mohl dlouhodobě trénink podstupovat a nevyhořel) a zároveň musí na maximum využívat sportovcův potenciál. Nalezení hrany, na které lze cvičit je ona stěžejní část. Pokud je trénink příliš pod ní, atlet nedosáhne na vrchol, pokud ji překročí, tak už vůbec ne. K zaznamenávání tréninků se běžně používala tužka a papír, dnes to jsou poznámky v elektronické podobě. Holý text ale není tak přehledný a těžko se z něj dají vyčíst důležitá data potřebná k naplňování dalších tréninků.

Mým cílem bylo sestavení webové aplikace se zobrazením tréninků a výkonů v co nejprehlednější podobě. Aplikace by měla obsahovat registraci i přihlášení uživatelů, a umožnit jim vytvoření svých vlastních typů tréninku a následné zaznamenávání cvičení. Cvičení by se měla zobrazovat v kalendáři, a to tak, aby šlo vyčíst co nejvíce informací. Další část aplikace by měla sloužit k zaznamenávání výkonů. Hodně funkcí by zde bylo podobných. Jediný podstatný rozdíl by byl v zobrazování dat, jelikož tady se jeví jako nejlepší možnost vizualizace výkonů v grafu. Celá webová stránka by měla být plně responzivní, aby fungovala na různých zařízeních.

V dokumentaci projektu podrobně řeším postup vytvoření dané aplikace. V úvodu se zabývám návrhem databázového modelu. Následně řeším uživatelské účty, formuláře pro přidávání a změnu dat a vytvoření kalendáře a grafů. V závěru pak samotnou vizuální stránku a její adaptivitu.

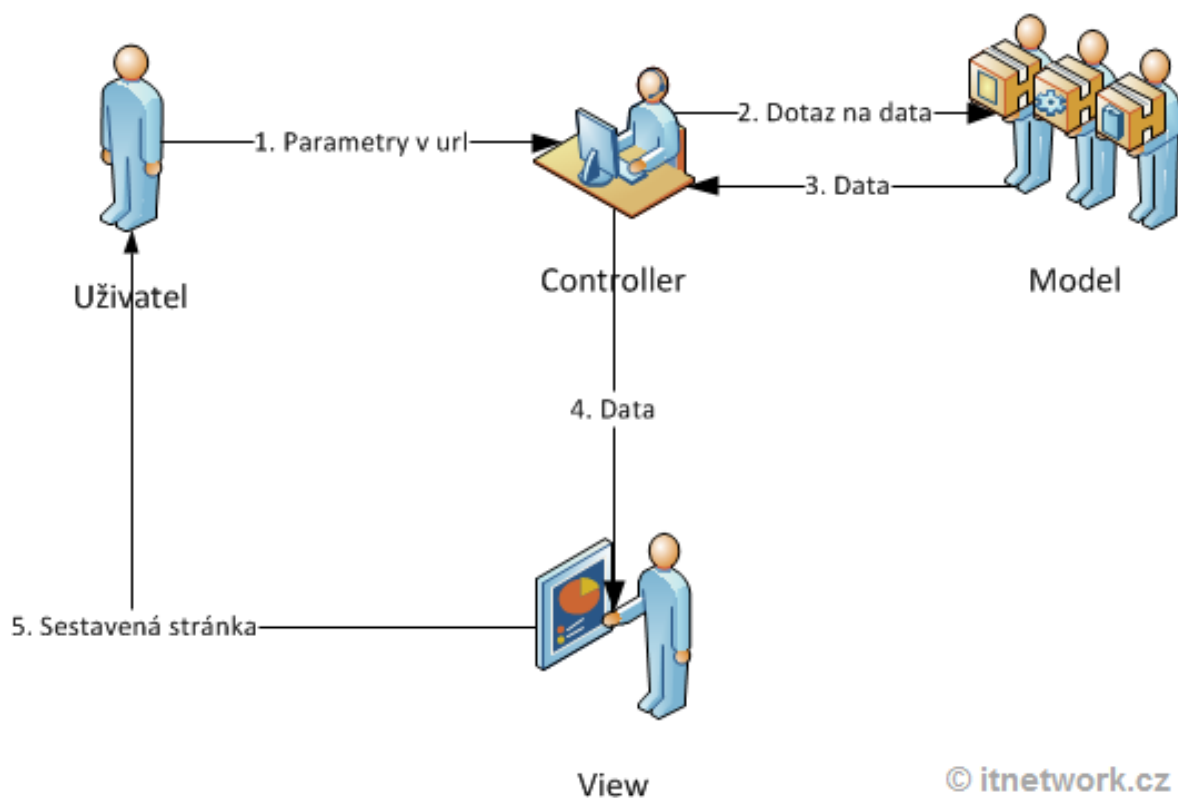
1 TEORETICKÁ A METODICKÁ VÝCHODISKA

1.1 Elektronické deníky a poznámky

Aplikace, ve kterých je možné zaznamenávat si svá cvičení, na internetu již existují. Těchto aplikací však není mnoho. Sám jsem takovou aplikaci kdysi hledal a většina z nich byla pro mě příliš složitá nebo nepřehledná. Složitost pochopení funkčnosti aplikace uživatele samozřejmě odrazuje. Zaznamenávání výkonů však nemusí být nijak sofistikované a může být velice přínosné i pro amatérské sportovce.

1.2 Architektura databázových aplikací

Databázové aplikace jsou většinou postavené na architektuře MVC. Ta dělí celou aplikaci na komponenty 3 typů, hovoříme o Modelech, View (pohledech) a Controllerech (kontrolérech), od toho MVC. Stručně nastíním cyklus, ve kterém tato architektura pracuje. Uživatel odešle požadavek a kontrolér jej zpracuje. Pokud uživatel požaduje data z databáze, kontrolér zavolá model a ten mu dané údaje vrátí. Nakonec řadič tyto data pošle do pohledu, respektive vygeneruje se šablona s daty. Hotová stránka je takto následně zobrazena uživateli.



Obrázek 1: Znázornění logiky MVC architektury (zdroj: itnetwork.cz)

1.3 Počáteční zkušenosti

Problematika databázových aplikací pro mě není zcela neznámá. Již jsem vytvořil jednu jednoduchou aplikaci v Django ve třetím ročníku. Využíval jsem u ní pouze generic-kých funkcí a uživatelský systém zde byl řešen holým Djangoem. Práci s databázemi samotnými jsem měl také osahanou z třetího ročníku, kde většina mých zkušeností vycházela z používání nástroje pro jednoduchou správu obsahu databáze MySQL prostřednictvím webového rozhraní phpMyAdmin.

2 VYUŽITÉ TECHNOLOGIE

2.1 Django CMS

Nejpopulárnější redakční systém založený na šablonovacím systému Django. Umožňuje tedy velice snadné přidávání nových elementů na stránku a jejich následnou správu. V mém projektu mi značně pomohl s vytvářením šablon.

2.2 PostgreSQL

PostgreSQL, často jednoduše Postgres, je svobodný a otevřený objektově-relační databázový systém. Funkce PostgreSQL zahrnují databázové transakce s atomicitou, konzistencí, izolovaností a trvalostí (ACID), automaticky aktualizovatelné pohledy, materializované pohledy, trigger, cizí klíče a uložené procedury.

Původně jsem pro tento projekt zamýšlel využít systém MySQL, avšak ten je pouze relační a neumožňuje tedy komplexnější datové typy či dědičnost. Funkce MySQL by pro můj projekt byly dostačující, ale s MySQL jsem již zkušenosti měl a chtěl jsem se přiučit jiným alternativám.

2.3 Bootstrap 5

Nejnovější verze Bootstrap má většinu funkcí stejnou jako Bootstrap 4, avšak celý framework je převeden na technologii Sass. Je to moderní preprocesor CSS, dovoluje použití proměnných, vrstvení selektorů nebo dědičnost. Pro implementaci Bootstrap 5 jsem se rozhodl hlavně díky těmto novým vlastnostem.

2.4 Knihovna jQuery

Tato JavaScriptová knihovna má jednoduchou syntaxi, výrazně zjednodušuje manipulaci s obsahem stránky, reakci na události, animace a používání AJAXu. V mém projektu jsem se ji rozhodl použít, protože už jsem s ní měl pár zkušeností.

2.5 Chart.js

Chart.js je open-source technologie umožňující generování grafů pomocí HTML Canvas. K dispozici nabízí několik druhů grafů a je funguje responzivně. Podobných technologií existuje mnoho. Rozhodl jsem se zrovna pro Chart.js, protože se mi jeho použití zdálo nejsnadnější.

2.6 Docker

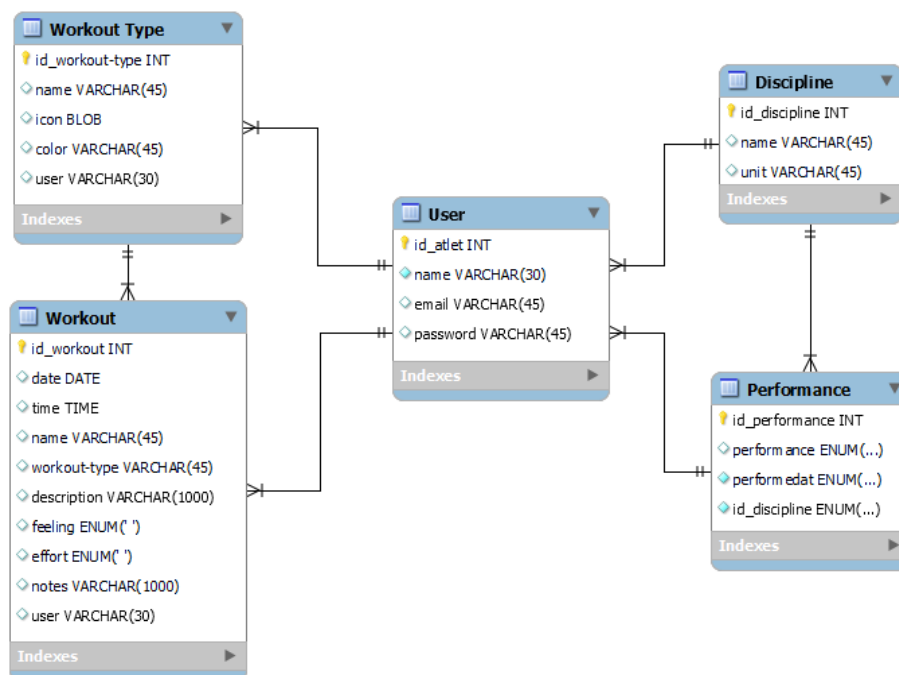
Docker je otevřený software (open source projekt), jehož cílem je poskytnout jednotné rozhraní pro izolaci aplikací do kontejnerů v prostředí macOS, Linuxu i Windows („odlehčená virtualizace“). Je široce využíván při vývoji jakéhokoliv softwaru a chtěl jsem s ním také nabít jisté zkušenosti.

3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

3.1 Založení projektu

Prvním krokem bylo stažení si Pythonu. Další klasický postup byl instalovat DjangoCMS. Od začátku projektu jsem ale věděl, že chci mít aplikaci zapouzdřenou v Dockeru. Tuto kombinaci technologií již jistě použily spousty vývojářů, poohlédl jsem se tedy po již sestaveném projektu, na němž bych mohl začít stavět vlastní aplikaci. Na GitHubu jsem našel oficiální repositář od společnosti Divio, který obsahuje jak DjangoCMS, tak i Docker soubory, tedy docker-compose.yml, potřebný pro konfiguraci Dockeru, a Dockerfile, zajišťující vytvoření Docker image. Naklonoval jsem jej a spustil pomocí přiložených instrukcí. Následně stačilo vytvořit v projektu vlastní aplikaci a měl jsem základní skeleton stránky a mohl se pustit do vytváření modelů.

3.2 Databázový model



Obrázek 2: ER diagram

3.3 Adresářová struktura

```
├──accounts                // aplikace accounts z balíčku django-allauth
├──data
├──log                    // vlastní aplikace
|  ├──migrations          // migrace
|  ├──static              // statické soubory aplikace
|  ├──templates           // adresář obsahuje HTML šablony
|  |  └──log
|  ├──admin.py            // registrace modelů
|  ├──urls.py             // cesty url
|  ├──settings.py         // nastavení aplikace
|  ├──models.py           // samotné modely
|  ├──views.py            // řadič
|  ├──forms.py            // formuláře
├──quickstart             // kořenová složka projektu
|  ├──templates           // kořenové šablony
|  ├──settings.py         // nastavení projektu
|  ├──urls.py             // cesty url
├──requirements.txt       // soubor s požadavky technologií
├──Dockerfile             // slouží pro vytvoření docker image
└──docker-compose.yml     // konfigurace dockeru
```

Obrázek 3: Adresářová struktura aplikace

3.4 Autentizace a autorizace

Uživatelský systém se stará o přihlašování a registrování uživatelů. Dále je důležité povolit uživatelům pouze akce, jež neporuší chod aplikace. Na začátku projektu jsem tuto problematiku vyřešil jednoduše pomocí holého Django, tak jako u všech mých předchozích projektů. Autentizační systém Django je ale velice strohý a neobsahuje příliš mnoho mož-

ností. Většinou se tudíž využívá externích balíčků, které ony rozšířenější požadavky umožňují. I já jsem se tedy rozhodl jeden z nich použít. Mou volbou se stal ten nejpopulárnější ze všech, a to django-allauth.

3.4.1 Balíček django-allauth

Zahrnuje registraci, přihlašování, základní operace s uživatelským účtem, a dokonce také autentizaci třetí strany. Nedílnou součástí jsou také jednotlivé šablony, které jsem si nasyloval pomocí Bootstrap. Pro stylizaci formulářů jsem použil balíček django-crispy-forms.

Pro uživatele proces autentizace probíhá zcela standardně jako na všech jiných webových stránkách. To samé platí pro operace s uživatelským účtem.

Balíček stačilo nainstalovat a zahrnout jej v souboru settings.py. Pro funkční posílání emailů (k ověření uživatelů, reset hesla atd.) jsem zde také nakonfiguroval nutné parametry. Django pro posílání mailů používá SMTP (Simple Mail Transfer Protocol), tedy standardní internetový protokol.

3.4.2 Autentizace pomocí aplikací třetích stran

Většina dnešních aplikací umožňuje uživateli autentizaci nejen pomocí emailu, ale také rychlejší cestu, a sice pomocí účtu, jenž už mají u jiné aplikace. Dvě největší a nejčastěji používané aplikace k tomuto účelu jsou bezesporu Google a Facebook.

V obou případech byl postup velice podobný. Přihlásil jsem se na stránky pro vývojáře a přidal vlastní aplikaci pro vytvoření API. Vygenerované ID a klíč jsem poté zadal v administrační části mé aplikace. Tento systém je založen na protokolu OAuth (Open Authorization), u kterého uživatelé přistupují k aplikacím bez hesla.

Sign Up

Already have an account? Then please [login](#).

Username*

E-mail (optional)

Password*

Password (again)*

Sign Up »

 GOOGLE

 FACEBOOK

Login

Please sign in with one of your existing third party accounts. Or, [sign up](#) for a example.com account and sign in below:

or

Username*

Password*

☐ Remember Me

[Forgot Password?](#)

Login

 GOOGLE

 FACEBOOK

Obrázek 4: Formuláře registrace a přihlášení

3.4.3 Základní operace s uživatelským účtem

Balíček django-allauth umožňuje uživatelům základní operace s jejich účtem. A i k těmto funkcím poskytuje příslušné šablony. Mezi operace patří:

- potvrzení emailu,
- změna emailu,
- reset hesla,
- změna hesla.

Aby k těmto funkcím měl uživatel snadný přístup, vytvořil jsem profilovou stránku, která odkazuje na podstránky řešící dané funkce, jelikož django-allauth takovou stránku sám o sobě neobsahuje.

Profile settings

Sign out

- [Change email](#)
- [Change password](#)
- [Forgotten password](#)

Obrázek 5: Nastavení účtu

3.5 Záznamy tréninků

Uživatel si bude moci vytvořit vlastní záznam o cvičení. Může u něj upravovat všechny parametry kromě parametru uživatel. Tam se totiž přiřadí jeho jméno. Aplikace pomocí tohoto parametru poté uživateli ukazuje z databáze pouze cvičení, které mu patří.

Důležitým parametrem cvičení je jeho druh. Je tím myšlen druh sportu či určitá disciplína. Uživatel má po přihlášení k dispozici několik základních druhů, avšak může si přidat i své vlastní. Druhy později slouží k filtraci cvičení.

Add a Workout Type

Name of the workout type*

Easy-paced run, Strength training,...

Color*

Save

Obrázek 6: Formulář pro přidání druhu tréninku

3.5.1 Kalendář

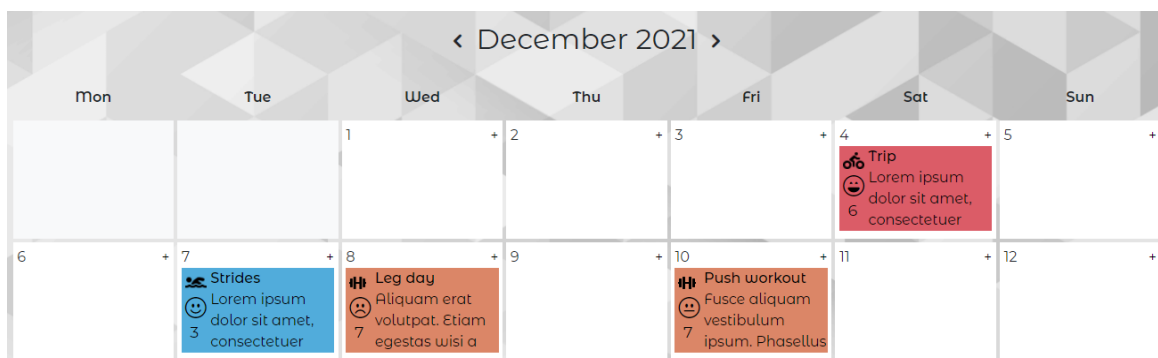
Přehledné zobrazení byla stěžejní část práce. Jako nejsrozumitelnější způsob výčtu cvičení se nakonec jevil měsíční kalendář.

Pro vytvoření kalendáře bylo hned několik způsobů. Mohl jsem jej celý vytvořit sám pomocí technologií jako jsou JavaScript a AJAX. Dělat všechno sám se ale zdálo zbytečné, jelikož na internetu mnoho vývojářů už řešilo tento úkol. Další možností bylo tedy použít nějaký již vytvořený balíček k tomuto účelu. Na internetu jsem našel pár kvalitních balíčků, avšak všechny byly příliš sofistikované a nelíbila se mi představa nemít celou věc pod kontrolou.

Nakonec jsem se tedy rozhodl pro v podstatě zlatou střední cestu. K vytvoření kalendáře použít Python modul HTMLCalendar. Využívá klasické Python funkce pro získání datumu a je navíc doplněn o funkce vracející zformátovaná data v podobě HTML tabulky. Jeho výhodou je také snadná úprava CSS nebo možnost přetížení jeho funkcí.

Při jeho implementaci jsem vycházel z oficiální dokumentace a z pár hotových příkladů, které tento modul používají. S grafickou úpravou jsem následně díky skvělým vlastnostem tohoto modulu neměl sebemenší potíže.

Díky kalendáři jsem poté přidal i velice dynamický způsob přidávání cvičení. Kliknutí na políčko dne v kalendáři pošle uživateli formulář pro vytvoření cvičení s již předvyplněným parametrem datum (datum dne, na který klikl).



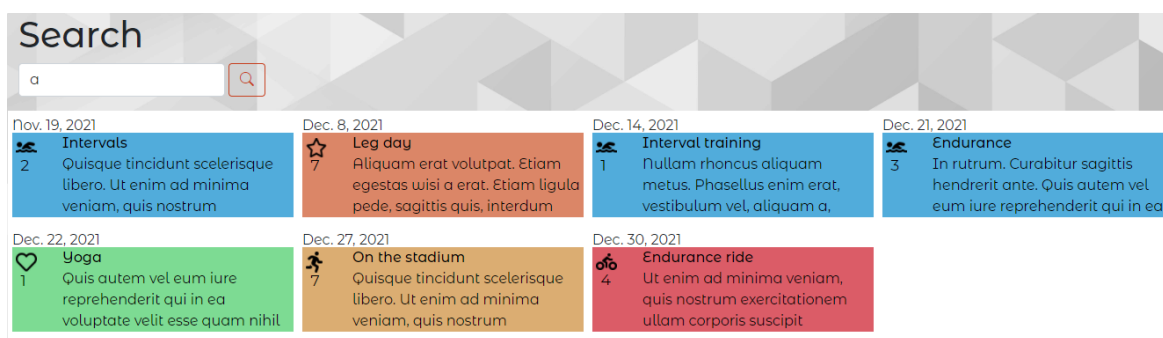
Obrázek 7: Kalendář tréninků

3.5.2 Filtrování

Uživatel má možnost filtrovat v kalendáři záznamy podle druhu cvičení. Tuto funkci jsem zajistil pomocí JavaScript knihovny jQuery, která práci velice zjednodušuje.

3.5.3 Vyhledávání

Nakonec jsem ještě chtěl uživateli udělit možnost vyhledávat ve svých cvičeních. Vytvořil jsem tedy vyhledávací formulář a nechal jej odkazovat na podstránku s výčtem cvičení, které odpovídaly zadanému textu. Formulář odešle hledaný výraz a řadič pošle pouze cvičení, které tento výraz obsahují.



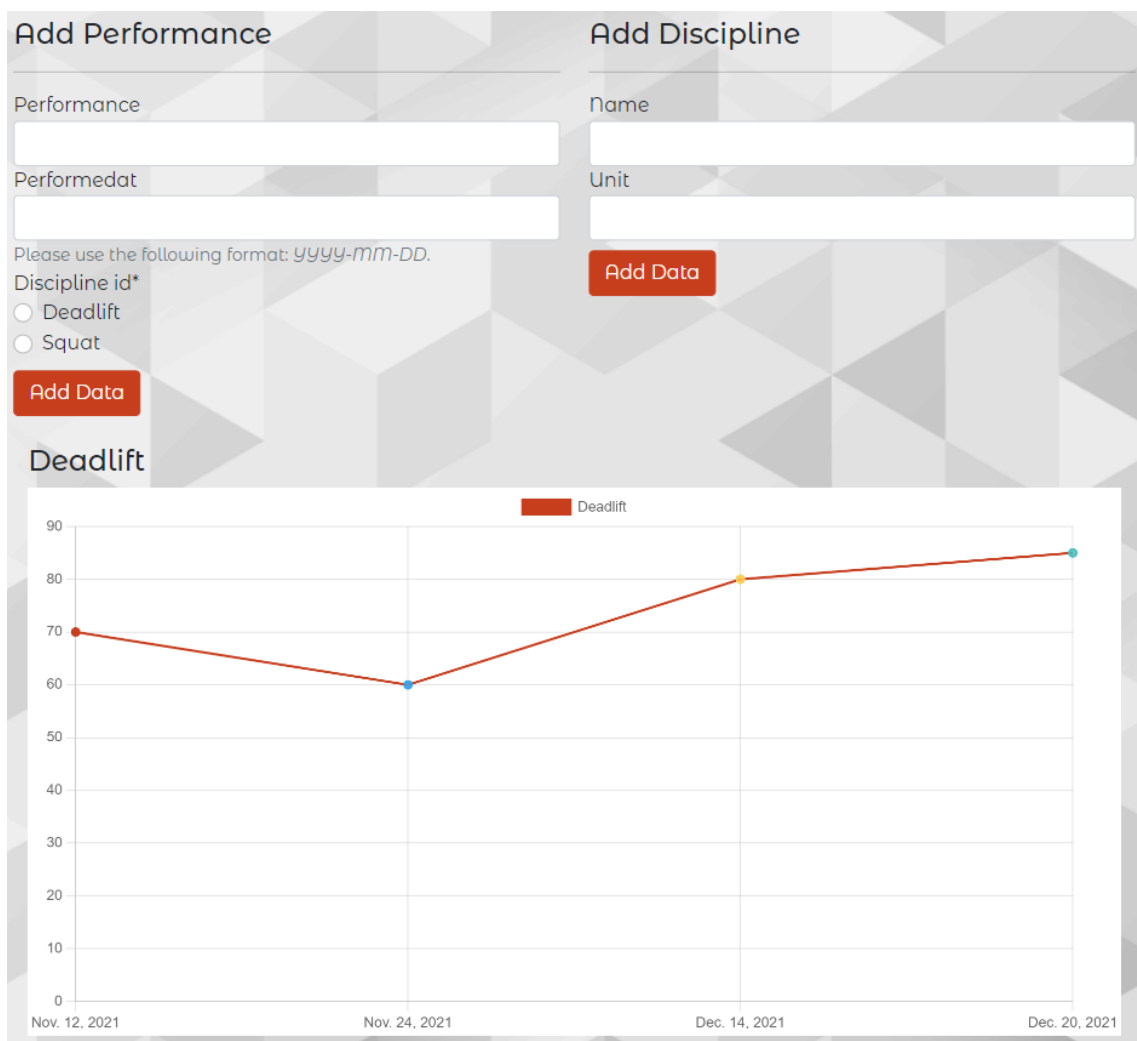
Obrázek 8: Vyhledávání cvičení

3.6 Záznamy výkonů

Další část aplikace má za úkol umožnit uživateli zaznamenávat své výkony a jejich následné zobrazení v grafu. Ze všeho nejdříve si uživatel vytvoří novou disciplínu s parametry jméno a jednotka. V dalším formuláři potom zaznamenává jednotlivé výkony s parametry disciplína, datum a výkon. Věděl jsem, že nejlepší způsob, jak tyto data zobrazit bude pomocí grafů. Existuje spousta technologií k tomuto účelu. Já se rozhodl pro JavaScriptový framework Chart.js.

Chart.js přijímá data v JSON podobě. Zpočátku jsem se zabýval tím, jak data zapsat do příslušného formátu (do hranatých a složených závorek). Řadič Django však sám umožňuje odesílání tohoto typu dat pomocí JsonResponse.

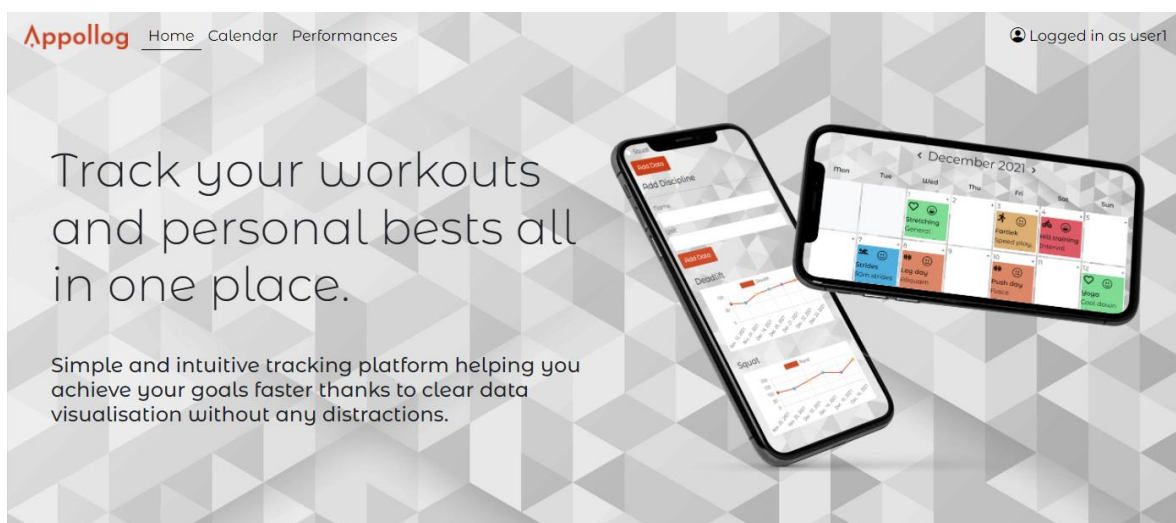
Protože jsem chtěl vypsát více grafů na jednu stránku, použil jsem v šabloně vnořené funkce for.



Obrázek 9: Graf výkonů

3.7 Úvodní strana

Na začátku úvodní stránky je stručný úvod do aplikace, sloužící návštěvníkovi k zorientování se, na jakou stránku vlastně přišel. Následuje vysvětlení benefitů, které přináší zaznamenávání tréninků a výkonů, aby potenciální uživatel pochopil výhody používání takové aplikace. Poté je stručné seznámení s funkcemi aplikace. Na konci stránky se nachází výzva k registraci, respektive tlačítko registrovat se.



Obrázek 10: Úvodní strana

4 VÝSLEDKY ŘEŠENÍ, VÝSTUPY, UŽIVATELSKÝ MANUÁL

4.1 Funkce aplikace

Při příchodu na stránku jsou uživateli nabídnuta tlačítka pro registraci či přihlášení. Svou autentizaci může provést také prostřednictvím Google a Facebook. Následně se mu otevře přístup ke všem stránkám. Na stránce kalendáře si může vytvářet svá cvičení s základními druhy cvičení nebo si vytvořit svůj vlastní. Stránka s výkony mu nabízí dva formuláře, jeden, pro vytváření disciplín, a druhý, pro zaznamenávání výkonů samotných. Z těchto dat se mu následně na stránce generují grafy. Podstránka profilu mu nabízí provést všechny základní operace se svým účtem jako jsou změna nebo reset hesla či změna emailu. Jeho data vidí pouze on sám a nemá možnost nijak narušit chod aplikace. Jinak řečeno, aplikace má pod kontrolou autorizaci. Aplikaci lze používat na zařízeních s jakoukoliv velikostí displeje, jelikož je responzivní. K funkcím aplikace jsem se snažil vytvořit logický přístup, aby používání aplikace bylo intuitivní.



Obrázek 11: Struktura podstránek

4.2 Splněné a nesplněné cíle

Cílem bylo vytvořit aplikaci a při jejím vytváření pochopit řadu principů a technologií. Většinu cílů jsem splnil a aplikace funguje. Kód však není napsán zcela optimálně a věřím, že by mohl být mnohem lepší a přehlednější. Do budoucna by se aplikace mohla dále rozvíjet. Také bych si přál ji v nejbližší době zveřejnit. Aplikaci je možno si nyní vyzkoušet pomocí naklonování mého GitHub repozitáře a provedením několika příkazů, které jsou u repozitáře připojeny. Díky Dockeru ji lze spustit na zařízení s jakýmkoli operačním systémem.

ZÁVĚR

Cílem projektu bylo vytvoření webové aplikace, umožňující uživatelům zaznamenávání tréninků a výkonů. Aplikace je postavená na frameworku Django, respektive na redakčním systému Django CMS. Uživatelský systém zajišťuje balíček django-allauth a front-end je řešen pomocí jQuery a Bootstrap 5. Na pozadí aplikace běží databázový systém PostgreSQL a je celá zapouzdřená v Dockeru.

Základem aplikace je kalendář, do kterého si uživatel zapisuje svá provedená cvičení. Druhá podstatná část slouží k zaznamenávání výkonů. Zde jsou data vizualizována pomocí grafů. Uživatel má možnost autentizace pomocí Google či Facebook. K dispozici má také standartní operace se svým účtem jako je změna hesla či potvrzení emailu. Uživatelský systém by mohl být uplatněn u téměř všech aplikací. Kalendář by mohl sloužit k zaznamenávání různých událostí a k sestavování programu k různým účelům. Část aplikace, která je zodpovědná za správu výkonů, by rovněž mohla být snadno přepracovaná k jiným účelům. Aplikace je zcela funkční, a to na všech zařízeních díky responzivnímu designu.

Aplikace je zálohovaná na GitHub adrese <https://github.com/lahodny/log-app>.

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- [1] Django: The web framework for perfectionists with deadlines. [online]. [cit. 2021-12-16]. Dostupné z: <https://www.djangoproject.com/>
- [2] Python: calendar — General calendar-related functions [online]. [cit. 2021-12-16]. Dostupné z: <https://docs.python.org/3/library/calendar.html>
- [3] Django Web Framework (Python). MDN Web Docs [online]. [cit. 2021-12-16]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Djang>
- [4] Bootstrap: The most popular HTML, CSS, and JavaScript framework for developing responsive, mobile first projects on the web. [online]. [cit. 2021-12-16]. Dostupné z: <https://getbootstrap.com/>
- [5] W3Schools: Online Web Tutorials [online]. [cit. 2021-12-16]. Dostupné z: <https://www.w3schools.com/>
- [6] Django CMS: Enterprise Content Management with Django [online]. [cit. 2021-12-16]. Dostupné z: <https://www.django-cms.org/>
- [7] Docker: Empowering App Development for Developers [online]. [cit. 2021-12-16]. Dostupné z: <https://www.docker.com/>
- [8] How to create a django CMS application with our quickstart repository [online]. [cit. 2021-12-16]. Dostupné z: <https://docs.divio.com/en/latest/how-to/quickstart-django-cms/>
- [9] Welcome to django-allauth!: django-allauth 0.43.0 documentation [online]. [cit. 2021-12-16]. Dostupné z: <https://django-allauth.readthedocs.io/>
- [10] Google Developers [online]. [cit. 2021-12-16]. Dostupné z: <https://developers.google.com/>
- [11] Facebook for Developers [online]. [cit. 2021-12-16]. Dostupné z: <https://developers.facebook.com/>

- [12] Forms have never been this crispy: django-crispy-forms 1.11.1 documentation [online]. [cit. 2021-12-16]. Dostupné z: <https://django-crispy-forms.readthedocs.io>
- [13] Google Fonts: Browse Fonts [online]. [cit. 2021-12-16]. Dostupné z: <https://fonts.google.com/>
- [14] Simple is Better Than Complex: Stories about Python, Django and Web Development [online]. [cit. 2021-12-16]. Dostupné z: simpleisbetterthancomplex.com
- [15] Django-colorfield: PyPI [online]. [cit. 2021-12-21]. Dostupné z: <https://pypi.org/project/django-colorfield/>