

ZÁVĚREČNÁ STUDIJNÍ PRÁCE dokumentace

Webová aplikace pro zaznamenávání tréninků
a výkonů (frontend část)



Autor: Barbora Žemličková
Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování
Třída: IT4
Školní rok: 2025/26

Poděkování

Ráda bych tímto poděkovala pánům učitelům Ing. Petru Grussmannovi a Mgr. Marku Lučnému za jejich cenné rady, připomínky a za pomoc s projektem.

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracovala samostatně a uvedla veškeré použité informační zdroje.

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

V Opavě 1. 1. 2026

.....
Podpis autora

Abstrakt

Výsledkem projektu je funkční webová aplikace Fit Track určená pro sledování fitness cílů a detailní analýzu tréninkového pokroku. Aplikace zahrnuje bezpečnou registraci a správu uživatelských účtů, včetně možnosti zjednodušeného přihlášení přes Google OAuth. Uživatel si v intuitivním rozhraní vede osobní tréninkový deník, kde eviduje jednotlivé cviky s parametry jako jsou série, opakování a váhy. Aplikace nabízí katalog cviků pro inspiraci a funkci, také "Rychlý start"s předpřipravenými plány pro různé úrovně pokročilosti. Dalsí částí pro zpětnou vazbu je analytický dashboard, který zpracovává data do interaktivních grafů. Ty vizualizují frekvenci cvičení, objem zvednuté zátěže a celkový progres v čase. Aplikace je postavena na frameworku Streamlit, který umožňuje vytvářet interaktivní datové aplikace v Pythonu s minimálním množstvím kódu a okamžitým náhledem v prohlížeči. Díky Streamlitu má Fit Track vícestránkové rozhraní – od tréninkového deníku přes katalog cviků až po analytický dashboard s grafy postavenými na knihovně Plotly, také disponuje moderním responzivním designem s tmavým motivem.

Klíčová slova

webová aplikace, fitness tracker, vizualizace dat, Python, Streamlit, tréninkový deník

Abstract

The result of the project is a functional web application Fit Track designed for tracking fitness goals and detailed analysis of training progress. The application includes secure registration and management of user accounts, including the option of simplified login via Google OAuth. The user keeps a personal training diary in an intuitive interface, where he records individual exercises with parameters such as sets, repetitions and weights. The application offers a catalog of exercises for inspiration and function, as well as a "Quick Start"with pre-prepared plans for different levels of progress. Another part for feedback is an analytical dashboard that processes data into interactive graphs. These visualize exercise frequency, volume of lifted load and overall progress over time. The application is built on the Streamlit framework, which allows you to create interactive data applications in Python with a minimum amount of code and an immediate preview in the browser. Thanks to Streamlit, Fit Track has a multi-page interface - from a training diary to an exercise catalog to an analytical dashboard with graphs built on the Plotly library. It also has a modern responsive design with a dark theme.

Keywords web application, fitness tracker, data visualization, Python, Streamlit, training diary

Obsah

1	Úvod	1
	Úvod	1
1.1	Zkušenosti	1
1.2	Splněné cíle	2
2	Využité technologie	3
2.1	Python	3
2.2	Streamlit	3
2.3	Plotly	3
2.4	Requests a JSON	4
2.5	Vývojové prostředí	4
3	Základní struktura	5
3.1	Struktura projektu ve Streamlitu	5
3.2	Komunikace s backend rozhraním (API)	5
3.3	Autentizace (Google OAuth)	6
4	Způsoby řešení a použité postupy	7
4.1	Založení projektu	7
4.2	Uživatelské rozhraní	7
4.3	Úvodní stránka	8
4.4	Hlavní přehled	9
4.5	Statistiky	10
4.6	Úspěchy	11
4.7	Katalog cviků	12
4.8	Záznamy tréninků	12
4.9	Autentizace a autorizace	13
5	Výsledky řešení a závěr	14
5.1	Výsledné řešení	14
5.2	Závěr	14
	Závěr	14

1 ÚVOD

Každý profesionální sportovec si vede řádný záznam o svých trénincích. Mít přehled v trénincích zajišťuje jejich lepší správu, a tedy i lepší výsledky. V dnešní době se k tomuto účelu již běžně nepoužívá tužka a papír, ale aplikace v elektronické podobě. Často se však jedná o ne-přehledné systémy, které neumožňují rychlou analýzu dat přímo v posilovně. Mým cílem bylo sestavení frontendové části webové aplikace Fit Track, která by v co nejpřehlednější podobě vizualizovala tréninkový progres. Aplikace je navržena tak, aby uživateli umožnila bezpečné přihlášení (včetně Google OAuth) a následné intuitivní vkládání výkonů. Dalsí částí mé práce je vytvoření analytického dashboardu, který surová data transformuje do interaktivních grafů. Celé rozhraní je plně responzivní s důrazem na tmavý motiv, který je pro fitness aplikace ideální. V dokumentaci se podrobně zabývám výběrem technologií pro frontend, realizací uživatelského rozhraní v prostředí Streamlit a propojením aplikace s backendovými službami.

1.1 ZKUŠENOSTI

Před začátkem práce na projektu Fit Track jsem neměla s frameworkm Streamlit žádné předchozí zkušenosti. Moje dosavadní znalosti webového vývoje se omezovaly na základy HTML a CSS, případně na jednoduché skripty v Pythonu. Tvorba komplexní webové aplikace s interaktivními prvky pro mě tedy byla zcela novou výzvou. Původně jsem zvažovala použití klasických cest (např. Flask nebo Django), ale brzy jsem narazila na vysokou časovou náročnost tvorby frontendu pomocí šablon a propojování s JavaScriptem. Jako začátečník v oblasti webových aplikací jsem hledala nástroj, který by mi umožnil rychle vytvořit profesionálně vypadající rozhraní bez nutnosti hluboké znalosti frontendových frameworků. Streamlit se ukázal jako ideální volba, i když vyžadoval zcela jiný způsob přemýšlení o tom, jak webová stránka funguje.

1.2 SPLNĚNÉ CÍLE

Splněné cíle:

- Vytvoření přehledného katalogu cviků: Systém umožnuje filtrování cviků podle svalových skupin pro snadnou orientaci.
- Možnost sestavovat tréninkové plány: Uživatel si může z katalogu vybrat konkrétní cviky a vytvořit si vlastní tréninkovou jednotku.
- Efektivní správa tréninků: Díky systému šablon a funkci „Rychlý start“ je proces záznamu tréninku maximálně zjednodušen.
- Přihlášení a autentizace: Zabezpečení přístupu je realizováno modulem auth.py, který zajišťuje ochranu uživatelských dat.
- Pokročilá analýza dat: Implementace interaktivních grafů v knihovně Plotly umožňuje detailní sledování progresu v sériích a opakování.
- Responzivní uživatelské rozhraní: Aplikace postavená na frameworku Streamlit je plně dostupná skrze webový prohlížeč na počítačích i mobilních zařízeních.
- Integrace s REST API backendu: Frontend pomocí knihovny Requests komunikuje s Flask serverem, načítá a ukládá tréninková data přes standardizované HTTP endpointy.
- Oddělené nasazení přes Docker Compose: Frontend má vlastní requirements.txt a je připraven běžet jako samostatná služba v Docker Compose, což usnadňuje nasazení a škálování.

Možné vylepšit:

- Detailní nahrávání videí.
- Mobilní rozhraní a nativní aplikace.

2 VYUŽITÉ TECHNOLOGIE

2.1 PYTHON

2.2 STREAMLIT

Streamlit je open-source framework určený pro rychlou tvorbu datových aplikací. Hlavním důvodem, proč jsem pro frontend zvolila Streamlit, je jeho úzká vazba na jazyk Python a také proto, že umožňuje definovat uživatelské rozhraní přímo v kódu bez nutnosti psát rozsáhlé HTML nebo CSS šablony. Streamlit se stará o vykreslování prvků a správu stavu aplikace.

2.2.1 Výhody využití Streamlitu

- **Extrémní rychlosť vývoje** – umožňuje vytvořit funkční rozhraní (tlačítka, grafy, nahrávání dat).
- **Čistě Pythonovské prostředí** – Nemusí se přepínat mezi Pythonem, HTML, CSS a JavaScriptem, vše se píše v jednom jazyce.
- **Nativní podpora pro interaktivní grafy** – skvěle si rozumí s knihovnami jako Plotly nebo Altair. Stačí mu předat data a on se postará o to, aby se graf správně zobrazil a byl interaktivní.
- **Hotové komponenty pro vstupy dat** – má v sobě už hotové prvky jako kalendáře, posuvníky nebo výběry z menu.
- **Interaktivita bez načítání** – ovládací prvky (tlačítka, posuvníky, výběry) přímo propojené s proměnnými v Pythonu.

2.3 PLOTLY

Pro generování grafů jsem využila knihovnu Plotly. Na rozdíl od statických obrázků umožňuje Plotly vytvářet interaktivní grafy, kde si uživatel může po najetí myší (nebo dotykem na mobilu) zobrazit přesné hodnoty jednotlivých sérií a tréninků.

2.4 REQUESTS A JSON

Jelikož je aplikace rozdělena na klientskou a serverovou část, využívám knihovnu Requests pro komunikaci s API. Data jsou přenášena ve formátu JSON, což zajišťuje lehkost a rychlosť přenosu informací mezi frontendem a databází.

2.5 VÝVOJOVÉ PROSTŘEDÍ

Jako hlavní nástroj pro psaní veškerého zdrojového kódu jsem zvolila editor Visual Studio Code. Na rozdíl od frameworku Streamlit, který byl pro mě novinkou, je pro mě toto prostředí velmi známé. VS Code používám jako svůj primární nástroj již od začátku studia na střední škole, což mi umožnilo se plně soustředit na řešení logiky aplikace a neřešit ovládání samotného editoru.

3 ZÁKLADNÍ STRUKTURA

Tato kapitola se zabývá základní strukturou frontendu webové aplikace Fit Track a popisuje hlavní části projektu, ze kterých se aplikace skládá.

3.1 STRUKTURA PROJEKTU VE STREAMLITU

- **static/** – adresář ve kterém jsou umístěny soubory css, pro tmavý režim stránky.
- **streamlit_app.py** – hlavní vstupní bod aplikace, tento skript vše odstartuje, stará se o úvodní nastavení stránky a propojuje jednotlivé moduly do jednoho celku
- **pages_dashboard.py** – logika pro vizualizaci tréninkových pokroků a generování interaktivních grafů, což je nejdůležitější část aplikace
- **pages_workouts.py** – správa tréninků, modul obsahující formuláře a rozhraní pro vkládání nových dat o cvičení, sériích a opakování
- **auth.py** – řeší logiku přihlašování uživatelů a zajišťuje, aby každý viděl pouze svá vlastní tréninková data
- **requirements.txt** – seznam závislostí, který je zcela zásadní pro spustitelnost aplikace; obsahuje přesný výčet Python knihoven (jako Streamlit nebo Plotly), na kterých projekt stojí
- **Dockerfile** – dokument, který umožňuje aplikaci "zabalit" do Dockeru, díky čemuž ji lze spustit na jakémkoliv počítači bez složitého instalování prostředí

3.2 KOMUNIKACE S BACKEND ROZHRANÍM (API)

Tato kapitola popisuje způsob, jakým frontendová část aplikace komunikuje s backendovým serverem, který vyvinul můj spolužák David Anděl, který pracoval na dokumentaci backendové části. Propojení obou částí probíhá pomocí architektury REST API.

3.2.1 Architektura propojení

Frontend funguje jako klient, který zasílá HTTP požadavky na specifické URL adresy (endpoints) backendu. Komunikace probíhá asynchronně, což zajišťuje, že se uživatelské rozhraní nezasekne během čekání na odpověď ze serveru.

3.2.2 Struktura API volání

V zájmu čistoty kódu a udržovatelnosti jsem v projektu implementoval(a) tzv. API Service vrstvu. To znamená, že HTTP požadavky nejsou rozesety přímo v komponentách, ale jsou soustředěny v samostatných modulech.

```
def check_login():
    """Check if user is logged in by calling /api/me"""
    session = st.session_state['session']
    try:
        r = session.get(f"{API_BASE}/me", timeout=2)
        if r.ok:
            st.session_state['logged_in'] = True
            st.session_state['user'] = _safe_json(r).get('user')
            return True
    except Exception:
        pass
    st.session_state['logged_in'] = False
    st.session_state['user'] = None
    return False
```

Ukázka ověření identity uživatele

3.3 AUTENTIZACE (GOOGLE OAUTH)

Jednou z klíčových funkcí frontendu je zjednodušené přihlášení. Využila jsem protokol OAuth 2.0, který uživateli umožňuje přihlásit se pomocí Google účtu. To zvyšuje bezpečnost (uživatel nemusí zadávat heslo do mé aplikace) a výrazně urychluje proces registrace.

Firestore umožňuje flexibilní ukládání dat a okamžitou synchronizaci, díky čemuž se změny projeví u trenéra i klienta bez nutnosti manuální obnovy stránky.

4 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

Tato kapitola popisuje způsob vytvoření frontendu aplikace Fit Track, základní postupy vývoje v prostředí Streamlit, funkční mechanismy využité při implementaci, včetně autentizace uživatelů a komunikace s backendem přes API.

4.1 ZALOŽENÍ PROJEKTU

Vývoj aplikace byl zahájen přípravou virtuálního prostředí v Pythonu a instalací frameworku Streamlit. Vzhledem k mým předchozím zkušenostem z výuky probíhal veškerý vývoj v editoru Visual Studio Code. Pro správu závislostí byl vytvořen soubor requirements.txt, který definuje potřebné knihovny, jako jsou Plotly pro grafy a Requests pro propojení s API. Celý frontend je také připraven pro kontejnerizaci pomocí přiloženého Dockerfile, což zajišťuje snadné spuštění aplikace v libovolném prostředí.

4.2 UŽIVATELSKÉ ROZHRANÍ

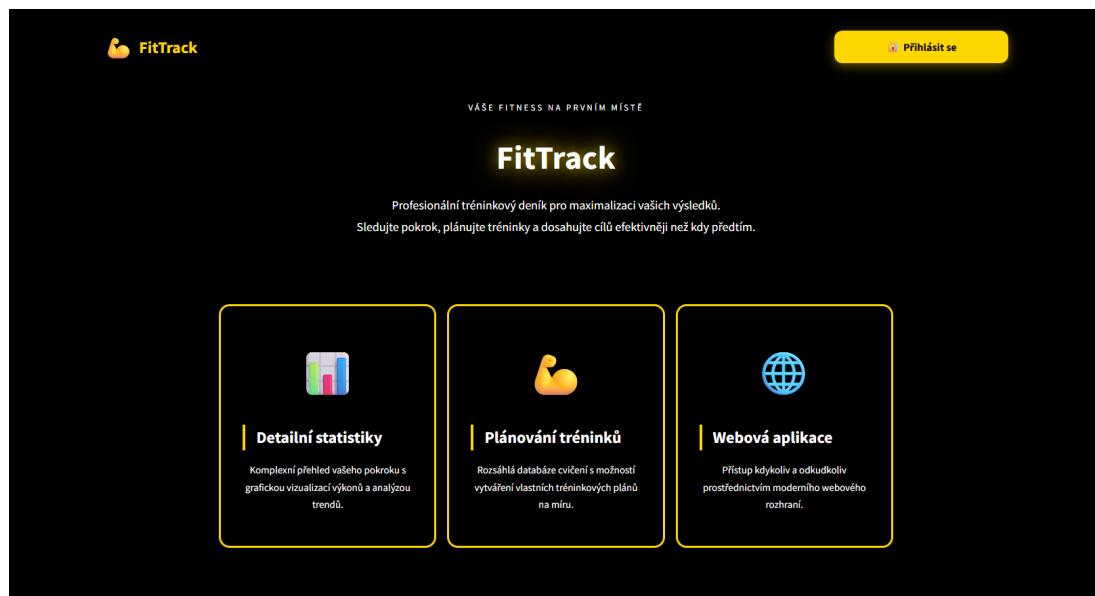
Uživatelské rozhraní aplikace je postaveno na frameworku Streamlit, který umožňuje skládat obrazovky z hotových komponent, jako jsou formuláře, tabulky, grafy nebo postranní panel. Aplikace využívá široké rozvržení s viditelným sidebar menu, odkud má uživatel rychlý přístup na hlavní sekce – přehledový dashboard, katalog cviků, tréninkový deník, analytické statistiky a nastavení účtu.

```
def render_sidebar_navigation():
    """Render sidebar navigation for logged-in users"""
    with st.sidebar:
        st.markdown("""
            <div style="text-align: center; padding: 20px 0;">
                <span style="font-size: 2.5rem;"></span><br>
                <span style="font-size: 1.8rem; font-weight: 800; color: #ffd700;">Fit
            </div>
        """, unsafe_allow_html=True)
```

```
st.markdown("---")  
  
# Navigation buttons  
current_page = st.session_state.get('page', 'dashboard')  
  
if st.button(" Dashboard", use_container_width=True, type="primary" if cu  
    st.session_state['page'] = 'dashboard'  
    st.rerun()  
  
if st.button(" Moje tréninky", use_container_width=True, type="primary" i  
    st.session_state['page'] = 'workouts'  
    st.rerun()
```

4.3 ÚVODNÍ STRÁNKA

Úvodní strana aplikace FitTrack je navržena s důrazem na moderní vzhled, přehlednost a okařitou srozumitelnost hlavních funkcí. Dominuje jí tmavý režim s výraznými žlutými akcenty, což odpovídá aktuálním trendům v oblasti fitness aplikací a zároveň snižuje únavu očí při používání v hűře osvětleném prostředí posiloven.

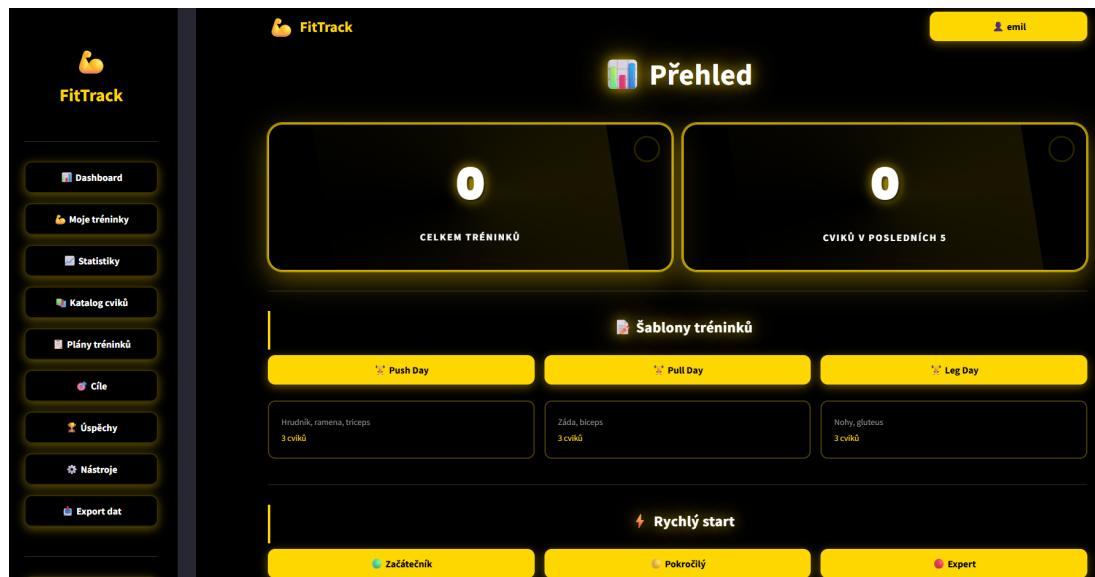


Obrázek 4.1: Hlavní stránka

4.4 HLAVNÍ PŘEHLED

Po úspěšném přihlášení je uživatel přesměrován na hlavní přehledovou stránku (Dashboard), která slouží jako centrální uzel aplikace. Tato část byla navržena tak, aby poskytovala okamžitou zpětnou vazbu o aktivitě uživatele bez nutnosti složitého vyhledávání.

- Navigační panel (Sidebar).
- Přehledové metriky.
- Šablony tréninků a Rychlý start.



Obrázek 4.2: Dashboard

4.5 STATISTIKY

Dalsí funkcí aplikace FitTrack je modul pro analýzu dat, který je implementován v souboru `pages_dashboard.py`. Cílem této části je poskytnout uživateli jasný přehled o jeho tréninkovém objemu a intenzitě pomocí interaktivních grafů.



Obrázek 4.3: Statistiky

4.5.1 Grafy

Pro vizualizaci jsem využila propojení Pythonu s knihovnou Plotly, což umožňuje generování responsivních grafů přímo v rozhraní Streamlit. V sekci „Analýza sérií a opakování“ aplikace zobrazuje dva klíčové pohledy:

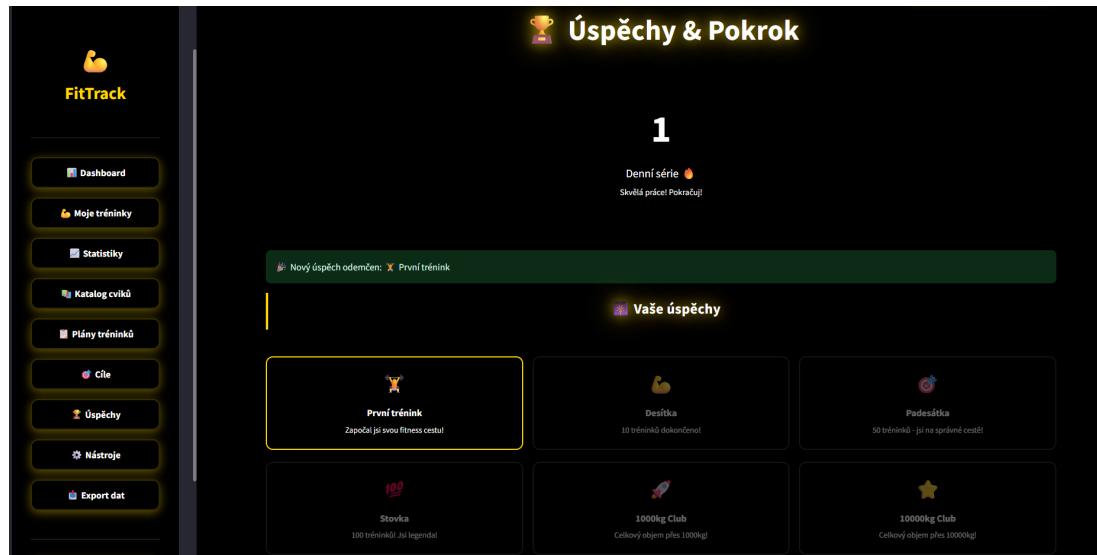
- Průměrný počet sérií (Top 10): Sloupcový graf zobrazuje průměrný objem sérií u nejčastěji vykonávaných cviků, jako jsou dřepy (Squats) nebo tlaky na lavici (Bench Press).
- Průměrný počet opakování (Top 10): Tento graf doplňuje předchozí pohled o informaci o průměrném počtu opakování, což pomáhá uživateli sledovat, zda se pohybuje v zóně pro budování sily nebo vytrvalosti.

4.5.2 Interaktivita a UX design

Grafy jsou plně interaktivní, což znamená, že uživatel může po najetí kurzorem na jednotlivé sloupce vidět přesné číselné hodnoty. Vizuální styl grafů byl upraven tak, aby doplňoval tmavý motiv aplikace – sloupce využívají modré odstíny, které jsou dobře čitelné na černém pozadí dashboardu.

4.6 ÚSPĚCHY

Důležitou součástí uživatelského zážitku v aplikaci FitTrack je systém úspěchů, který je implementován jako samostatný modul přístupný z navigačního panelu. Tento systém sleduje aktivitu uživatele a automaticky odemyká odměny na základě dosažených milníků.

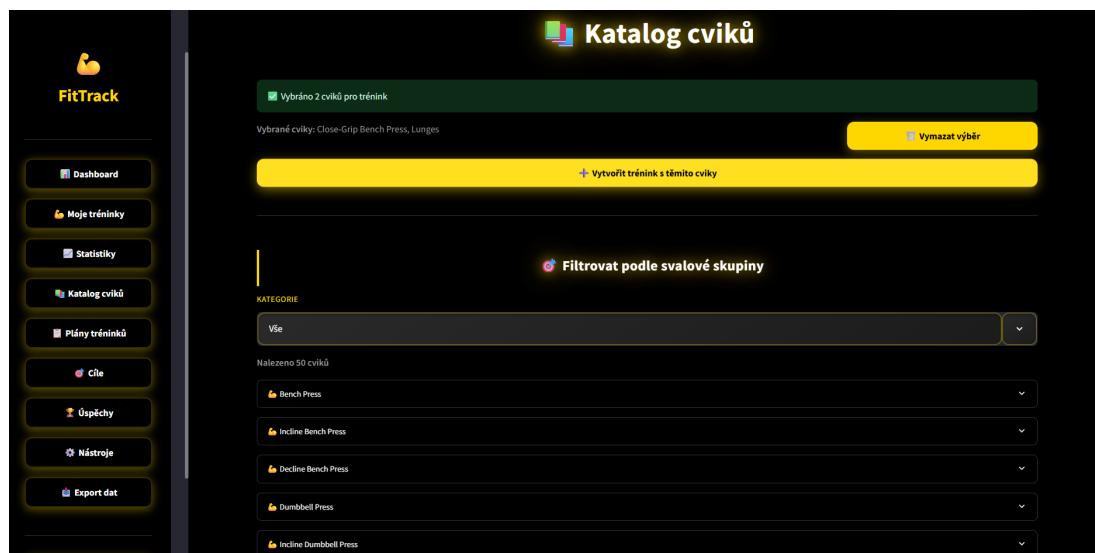


Obrázek 4.4: Úspěchy

4.7 KATALOG CVIKŮ

Jednou z klíčových komponent pro usnadnění uživatelské zkušenosti je Katalog cviků, který slouží jako rozsáhlá databáze znalostí v rámci aplikace. Tato část umožňuje uživatelům nejen prohlížet dostupné cviky, ale také aktivně sestavovat vlastní tréninkové jednotky.

- Multiselect výběr.
- Akční tlačítka.
- Správa výběru.



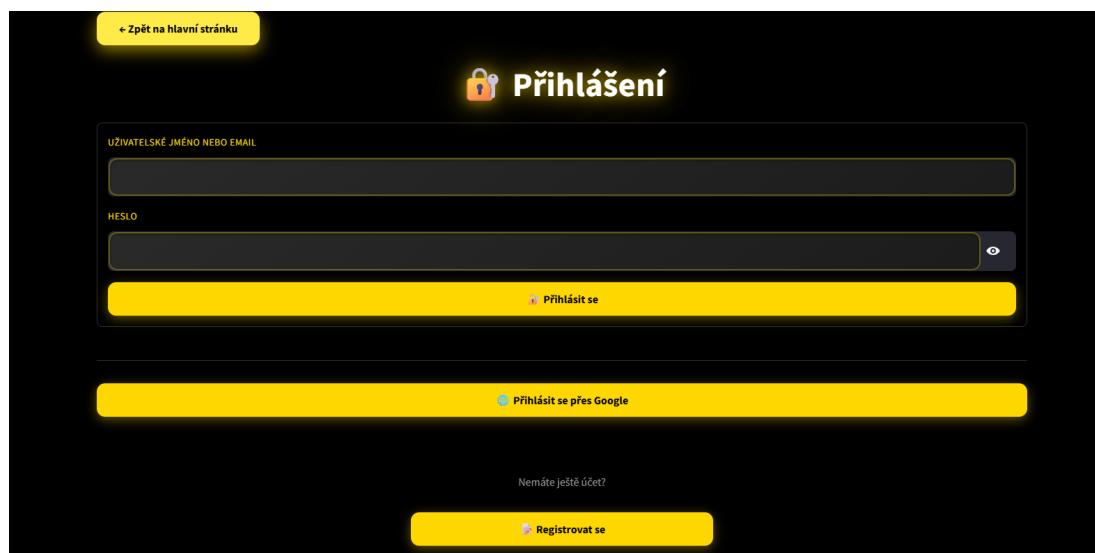
Obrázek 4.5: Katalog

4.8 ZÁZNAMY TRÉNINKŮ

Uživatel může vytvářet nové záznamy, ukládat jejich průběh v reálném čase a zpětně k nim přistupovat skrze historii. Pro maximální efektivitu v posilovně jsem implementovala systém šablon (např. Push Day, Pull Day), které umožňují rychlý start tréninku bez nutnosti ručního zadávání každého cviku. Každý záznam tréninku v databázi obsahuje detailní informace o vykonaných cvicích, včetně počtu sérií, opakování a použité zátěže.

4.9 AUTENTIZACE A AUTORIZACE

Autentizace uživatelů je implementována v modulu auth.py. Aplikace umožňuje uživatelům bezpečné přihlášení, přičemž je kladen důraz na ochranu dat – každý sportovec má přístup pouze k vlastním záznamům. Po úspěšném ověření jsou přihlašovací údaje a tokeny uloženy v rámci .session_state, což umožňuje aplikaci udržet uživatele přihlášeného i při pohybu mezi jednotlivými podstránkami, jako je deník tréninků nebo statistiky.



Obrázek 4.6: Proces autentizace uživatele v aplikaci Fit Track

5 VÝSLEDKY ŘEŠENÍ A ZÁVĚR

5.1 VÝSLEDNÉ ŘEŠENÍ

Výsledkem práce je funkční webová aplikace FitTrack, která slouží jako moderní tréninkový deník pro záznam a analýzu fitness aktivit.

5.2 ZÁVĚR

Cílem této závěrečné práce bylo navrhnout a realizovat webovou aplikaci FitTrack, která poskytuje moderní a přehledné prostředí pro digitální správu tréninkového deníku a analýzu sportovních výkonů. Na základě provedené analýzy potřeb sportovců a následného návrhu v prostředí Visual Studio Code vznikla aplikace postavená na jazyce Python a frameworku Streamlit.

Výsledné řešení splňuje většinu stanovených požadavků, zejména v oblasti interaktivní vizualizace dat pomocí knihovny Plotly, správy katalogu cviků a motivačního systému úspěchů. I přes určitá omezení v oblasti nativní mobilní platformy představuje FitTrack plně funkční a stabilní základ, který je díky své modulární struktuře a kontejnerizaci skrze Dockerfile připraven pro další rozšiřování a budoucí rozvoj.

LITERATURA

- [1] *Streamlit Documentation*.Main Concepts of Streamlit.[online]. Dostupné z: <https://docs.streamlit.io> [cit. 2025-09-25].
- [2] *Python Software Foundation*.Python Language Reference.[online]. Dostupné z: <https://www.python.org/doc/> [cit. 2025-09-30].
- [3] *Plotly*.Plotly Python Graphing Library.[online]. Dostupné z: <https://plotly.com/python/> [cit. 2025-10-08].
- [4] *Streamlit Docs*.Session State and Data Persistence.[online]. Dostupné z: https://docs.streamlit.io/develop/api-reference/status/st.session_state [cit. 2025-10-20].
- [5] *Docker Documentation*.Dockerize a Python application.[online]. Dostupné z: <https://docs.docker.com/language/python/> [cit. 2025-10-28].
- [6] Microsoft. *Visual Studio Code*.Python Development Guide.[online]. Dostupné z: <https://code.visualstudio.com/docs/languages/python> [cit. 2025-11-16].
- [7] *Streamlit API*.Theming and Layout Customization.[online]. Dostupné z: <https://docs.streamlit.io/develop/api-reference/configuration/config.toml> [cit. 2025-11-22].
- [8] *Plotly Docs*.Bar Charts and Statistical Visualizations in Python.[online]. Dostupné z: <https://plotly.com/python/bar-charts/> [cit. 2025-12-02].
- [9] *Python Pip*.Managing Dependencies with requirements.txt.[online]. Dostupné z: https://pip.pypa.io/en/stable/user_guide/ [cit. 2025-12-13].
- [10] *Docker Docs*.Best practices for writing Dockerfiles.[online]. Dostupné z: https://docs.docker.com/develop/develop-images/dockerfile_best-practices/ [cit. 2025-12-26].

Seznam obrázků

4.1	Hlavní stránka	8
4.2	Dashboard	9
4.3	Statistiky	10
4.4	Úspěchy	11
4.5	Katalog	12
4.6	Proces autentizace uživatele v aplikaci Fit Track	13

Výsledná aplikace zde: <https://github.com/Bara2222/fittrack>