

Gymnázium, Praha 6, Arabská 14

Obor Programování

ROČNÍKOVÁ PRÁCE



Prohlášení

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené.

Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

Obsah

Prohlášení.....	2
Anotace	4
Abstract	4
Zadání	5
1. Úvod.....	6
2. Použité technologie	6
2.1. Visual Studio Code.....	6
2.2. Python.....	7
2.3. Django	7
2.4. Django templates.....	8
2.5. Bootstrap 5	8
2.6. Rest Framework	9
3. Backend	10
3.1. Modely	10
3.2. Formuláře	11
3.3. Serializéry.....	12
3.4. Funkce	13
4. Frontend	15
5. Mimořádné chyby	17
6. Instalace.....	18
7. Závěr.....	19
Zdroje	20
Seznam obrázků.....	22

Anotace

Trénink Deník je webová aplikace sloužící k zapisování fyzických aktivit. Uživatel tak má možnost mít přehled o svém zdraví a fyzické aktivitě v průběhu každého měsíce. Má-li uživatel v oblibě týmové sporty, kondiční tréninky anebo posílení svalů v posilovně, vše se dá zaznamenat do Trénink Deníku.

Abstract

Training Diary is a web application used to record physical activities. This allows the user to keep track of their health and physical activity throughout the month. If the user enjoys team sports, intensity training, or strengthens their muscles at the gym, everything can be recorded in Training Diary.

Zadání

Zadáním bylo vytvořit aplikaci, která umožní uživateli zapisovat si svou denní aktivitu při sportu. Jeho vzhled by se podobal kalendáři, kam by se tréninky zapisovaly. Uživatel bude mít možnost zadat své tělesné parametry a za určitou dobu by uživatel zadal novější, což by se postupně vykreslilo do grafů. Bylo by na výběr z vícero druhů tréninku, podle nichž bude aplikace uživateli doporučovat náplň tréninku ideální pro jeho aktuální stav. Dodatečně může aplikace doporučovat i stravu.

1. Úvod

Aplikace Trénink Deník slouží k zapisování tréninků či jiných denních aktivit a monitorování svých tělesných hodnot. Zapisování tréninků je pro mě každodenním standardem, protože já sám se sportu věnuji. V mladším věku jsem dokonce obdržel i papírový deník pro stejné účely jako tato aplikace. Můj vztah ke sportu mě dostatečně motivoval k volbě tohoto tématu. Dokumentace obsahuje popis jednotlivých sekcí projektu a vysvětlení, jak celá aplikace funguje. Komplexnější části projektu jsou tak uvedeny dříve, aby byly vysvětleny a následně využity při rozboru ostatních oblastí.

2. Použité technologie

Během vývoje aplikace bylo využito mnoho přístrojů a doplňků, včetně knihoven. Pro vývoj aplikace jsem použil Visual Studio Code (*Visual Studio Code*, str. 6) podporující GitHub. Aplikace byla napsána v programovacím jazyce Python spolu s jeho webovým frameworkem Django (*Python & Django*, str. 7).

2.1. Visual Studio Code

Visual Studio Code je editor zdrojového kódu vyvíjený společností Microsoft pro operační systémy Windows, Linux a macOS. Obsahuje podporu pro Git (a pro GitHub), zvýraznění syntaxe, kontextový našeptávač a podporu pro ladění a refaktorizaci. Zdrojový kód je svobodný software pod licencí MIT. Sestavené verze nabízené přímo Microsoftem jsou freewarem obsahujícím telemetrii, ale existuje i komunitně

sestavovaná varianta VSCodium. Editor je naprogramovaný v JavaScriptu a TypeScriptu. [1]



Obr. 1. - logo Visual Studio Code

2.2. Python

Python je vysokoúrovňový, interpretovaný programovací jazyk, který v roce 1991 navrhl Guido van Rossum. Nabízí dynamickou kontrolu datových typů a podporuje různá programovací paradigmaty, včetně objektově orientovaného, imperativního nebo funkcionálního. Jako instalátor balíčků slouží např. pip, který je přítomný ve většině instalacích Pythonu. Tyto balíčky lze získat např. z repozitáře PyPI (Python Package Index). V roce 2023 se jednalo o třetí nejpoužívanější programovací jazyk. [2]

2.3. Django

Django je vysoko úrovní webový framework Pythonu, který podporuje rychlý vývoj a čistý, pragmatický design. Vytvořili jej zkušení vývojáři a postará se o většinu starostí s vývojem webu, takže se můžete soustředit na

psaní své aplikace, aniž byste museli znovu objevovat kolo. Django je zdarma a open source. [3]



Obr. 2. - logo frameworku Django

2.4. Django templates

Django templates jsou technologie, které určují strukturu, styl a chování webových stránek nebo aplikací. Jako webový framework přináší Django pohodlný způsob, jak dynamicky generovat HTML. [4]

2.5. Bootstrap 5

Bootstrap 5 je nejnovější verze Bootstrapu, což je nejpopulárnější rámec HTML, CSS a JavaScript pro vytváření responzivních webových stránek zaměřených na mobily. Bootstrap 5 je zcela zdarma ke stažení a použití. [5]

2.6. Rest Framework

Rest Framework je rozhraní pro programování aplikací (Application Programming Interface, zkráceně API), který je v souladu s principy návrhu architektonického stylu reprezentativního přenosu stavu. [6]

3. Backend

Backend byl napsán v Pythonu a slouží k celkovému chodu aplikace. Zahrnuje správu a zpracovává data v zákulisí. V dokumentaci jsou vyjmenovány formuláře, serializéry, modely a konkrétní funkce nezbytně důležité pro chod aplikace. [7]

3.1. Modely

Model uživatele (*Uzivatel*) obsahuje informace o registrovaném uživateli v aplikaci. Model je rozšířen o Django model *AbstractUser*, takže obsahuje základní autentizační funkce (heslo, přihlášení). Po registraci se každému uživateli automaticky vygeneruje identifikátor (*id*). Uživateli náleží jeho uživatelské jméno (*username*), e-mail (*email*) a pakliže uživatel zadá své údaje, model si přivlastní i jeho věk (*vek*), váhu (*vaha*) a výšku (*vyska*).

Model tréninku (*Trenink*) obsahuje informace o každém ze zapsaných tréninků. Model *Trenink* vlastní hlavní pole *user*, zapsán jako cizí klíč neboli *ForeignKey*, což zařídí, že každý trénink patří konkrétním uživatelům. Mezi další pole patří: *nazev* (název tréninku), *datum*, *type* (výběr z druhů tréninků), *doba* (uplynulý čas tréninku) a *pozn* (poznámky).

3.2. Formuláře

Formulář *RegistraceUseraForm* se používá pro registraci nového uživatele. Zobrazuje textová pole uživatelského jména, e-mailu a hesla. Formuláři náleží i dodatečná funkce, která zkontroluje, zda zadané uživatelské jméno již existuje.

```
# Formulář pro registraci uživatele
class RegistraceUseraForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput)

    class Meta:
        model = Uzivatel
        fields = ["username", "email", "password"]
        labels = {
            'username': 'Uživatelské jméno',
            'email': 'E-mail',
            'password': 'Heslo'
        }

    # Kontrola jména uživatele
    def clean_username(self):
        dejmijmeno = self.cleaned_data.get('username')
        if Uzivatel.objects.filter(username=dejmijmeno).exists():
            raise ValidationError("Uživatelské jméno již existuje. Zvolte jiné.")
        return dejmijmeno
```

Obr. 3. – kód formuláře *RegistraceUseraForm*

Formulář *UzivatelForm* slouží k zadání nebo úpravě uživatelských údajů. Dokud uživatel nevyplní tento formulář, jsou pole *vek*, *vaha* a *vyska* modelu *Uzivatel* prázdné.

Formulář *TreninkForm* se zobrazuje při zadávání tréninků. Do formuláře uživatel vyplňuje všechny pole modelu *Trenink*, vyjma pole *datum*, protože aplikace pracuje s datumem ve tvaru *YYYY-MM-DD*. Formulář se dá

se vyplnit i víckrát pro daný den a pokaždé je pole *datum* neměnitelný, aby se příslušné tréninky ukládaly do daného dne.

```
# Formulář pro zapsání tréninku
class TreninkForm(forms.ModelForm):
    class Meta:
        model = Trenink
        fields = ['nazev', 'datum', 'type', 'doba', 'pozn']
        labels = {
            'nazev': 'Název',
            'datum': 'Datum',
            'type': 'Typ tréninku',
            'doba': 'Doba (minuty)',
            'pozn': 'Poznámky'
        }

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['datum'].widget.attrs['readonly'] = True
        self.fields['pozn'].widget.attrs.update({'rows': 3, 'cols': 30, 'style': 'resize: none;'})
```

Obr. 4. – kód formuláře *TreninkForm*

3.3. Serializéry

Serializér uživatele (*UzivatelSerializer*) pracuje s uživatelskými daty a serializér tréninků (*TreninkSerializer*) pracuje s tréninky uživatelů. Příslušný serializér převádí všechny atributy modelů *Uzivatel* a *Trenink* do JSON formátu a zpět, což serializéru umožní přenos dat mezi backendem a frontendem anebo validaci dat. [9]

3.4. Funkce

Funkce *register* zobrazí formulář *RegistraceUseraForm*, po jehož vyplnění zkontroluje, zda uživatelské jméno již existuje. Jestli ne, uživatele si uloží natrvalo. Také ho ihned přesměruje k samotnému kalendáři.

Funkce *kalendar* generuje kalendář aktuálního měsíce, včetně správně seřazených dnů v týdnu. Samotné dny jsou v kalendáři zapsány ve formátu *YYYY-MM-DD* pro následné generování zápisu tréninku pro vybraný den. Vytváří proměnné předchozího a následujícího roku a měsíce pro možné přepínání mezi měsíci.

```
# Kalendář
cal = calendar.Calendar()
kalendar = []

treninky = Trenink.objects.filter(user=uzivatel, datum_year=rok, datum_month=mesic)
udelaltrenink = set(treninky.values_list('datum', flat=True))
for tyden in cal.monthdayscalendar(rok, mesic):
    tyden_dny = []
    for den in tyden:
        if den == 0:
            tyden_dny.append(None) # Prázdné místo pro dny předchozího a následujícího měsíce
        else:
            tyden_dny.append(f"{rok}-{mesic:02d}-{den:02d}") # Formát YYYY-MM-DD
    kalendar.append(tyden_dny)
```

Obr. 5. – kód tvořící kalendář

Funkce *uzivatel_udaje* zobrazí formulář *UzivatelForm*, kam si uživatel může zapsat své tělesné parametry. Pakliže jsou údaje vyplněny, uživatel si je může změnit, pokud chce.

Funkce *zapistreninku* zobrazí formulář *TreninkForm* pro zapisování tréninků. Tento formulář lze vyplnit tolikrát, kolik uživatel vykonal tréninků, nicméně neslouží k úpravě.

```
@login_required
def zapistreninku(request, datum):
    try:
        datum_date = datetime.strptime(datum, '%Y-%m-%d').date()
    except ValueError:
        return redirect('kalendar')

    # Filtrujeme tréninky pouze pro přihlášeného uživatele!
    treninky = Trenink.objects.filter(datum=datum_date, user=request.user)

    if request.method == "POST":
        form = TreninkForm(request.POST)
        if form.is_valid():
            trenink = form.save(commit=False)
            trenink.user = request.user # Přidá usera k tréninku
            trenink.datum = datum_date
            trenink.save()
            return redirect('zapistreninku', datum=datum_date)
    else:
        form = TreninkForm(initial={'datum': datum_date})

    return render(request, 'zapistreninku.html', {'datum': datum_date, 'treninky': treninky, 'form': form})
```

Obr.6. – funkce *zapistreninku*

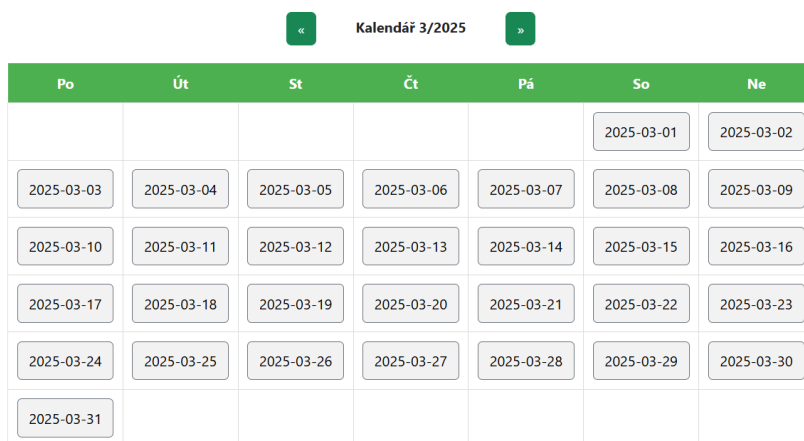
Funkce *treninky* vezme všechny tréninky zapsané v aplikaci a vyfiltruje pouze ty, které patří registrovanému uživateli.

4. Frontend

Frontend aplikace byl napsán v Django templates. Frontend zobrazuje data z backendu, reaguje na veškeré akce provedené uživatelem a přizpůsobuje vzhled. V dokumentaci jsou vyjmenovány HTML stránky tvořící samotnou webovou aplikaci. [7][10]

Stránky jsou pojmenovány stejně jako funkce, které je obsluhují a zajišťují jejich správné fungování. Analogicky tak stránka *register* zobrazí formulář *RegistraceUseraForm* anebo stránka *uzivatel_udaje* zase formulář *UzivatelForm*.

Stránka *kalendar* ukazuje aktuální měsíc rozdělený do týdnů vyplněné buňkami každého dne. Kliknutím na buňku bude uživatel přesměrován na stránku *zapistreninku*, kde mu stejnojmenná funkce zobrazí zápisník tréninku odpovídajícímu dni. Na stránce uživatel uvidí navíc i seznam tréninků zapsané pro tento den. Pokud uživatel spustí aplikaci znovu, aplikace ho přesměruje ihned ke kalendáři. [11][12]



Kalendář 3/2025						
Po	Út	St	Čt	Pá	So	Ne
					2025-03-01	2025-03-02
2025-03-03	2025-03-04	2025-03-05	2025-03-06	2025-03-07	2025-03-08	2025-03-09
2025-03-10	2025-03-11	2025-03-12	2025-03-13	2025-03-14	2025-03-15	2025-03-16
2025-03-17	2025-03-18	2025-03-19	2025-03-20	2025-03-21	2025-03-22	2025-03-23
2025-03-24	2025-03-25	2025-03-26	2025-03-27	2025-03-28	2025-03-29	2025-03-30
2025-03-31						

Obr. 7. – vzhled kalendáře

Na stránce *treninky* jsou jednotlivé tréninky přihlášeného zobrazeny do tabulky, jejíž sloupce odpovídají atributům modelu *Trenink* předtím zapsané skrze formulář *TreninkForm*. Uživatel si musí promyslet, co považuje za trénink, jelikož tréninky nelze upravovat ani odstranit.

Moje tréninky

Název	Datum	Typ tréninku	Doba	Poznámky
Cyklo na Ladronce	March 27, 2025	cyklo	228 min	Celodenní výlet s rodinou
Čau Pétó	March 28, 2025	workout	74 min	Potkal jsem Pétu! Jupí!
2R, tempáč 4km (tempo 10km), 2V	March 29, 2025	běh	43 min	

Obr. 8. – struktura seznamu tréninků

5. Mimořádné chyby

Během vývoje na Trénink Deníku jsem se potýkal s velkým množstvím chyb. Jeden z prvních problémů bylo samotné registrování uživatele, které způsobovalo chyby, kvůli nesprávně napsaným funkcím a špatně zadaným proměnným v modulech. Obě fixace problém opravily. Spolu s registrací aplikace neuměla ukládat registrovaného uživatele. Na místo komplexnějších funkcí jsem problém vyřešil přidáním lhůty vypršení registrace na neurčitou. Anebo sdílené tréninky se všemi uživateli čili stránka *treninky* zobrazovala všechny tréninky uložené v aplikaci. Lehkou opravou stejnojmenné funkce se problém vyřešil. [13][14]

6. Instalace

Když byla aplikace představena a popsána, je taktéž v dokumentaci i připsaný návod na instalaci Trénink Deníku. Celý postup instalace je uveden na obrázku s příkazy níže (*Obr. 9.*). Pakliže jsou všechny kroky splněny bez jakýkoliv komplikací, aplikace bude spuštěna.

```
git clone https://github.com/gyarab/2024-4e-pesula-TreninkDenik
cd treninkdenik
python -m venv venv
source venv/bin/activate
pip install -r requirements.txt
python manage.py migrate
python manage.py runserver
```

Obr. 9. – seřazený seznam příkazů pro instalaci aplikace [8]

Při instalaci se mohou nastat potíže. Na začátku je nutné vstoupit do adresáře *treninkdenik*, aby následující příkazy fungovaly. Pokud jsou potíže s instalací potřebných nástrojů a knihoven, je možné, že nebylo spuštěno virtuální prostředí či ani nebylo vytvořeno. Vytvořením a aktivováním virtuálního prostředí se můžou provádět další operace. Taktéž je třeba nainstalovat potřebné knihovny a nástroje, jinak aplikace nebude pracovat správně.

7. Závěr

Ročníkový projekt považuju za částečně dokončený. Podařilo se mi vytvořit webovou aplikaci, na kterou se uživatel může registrovat. Tréninky dopadly taky podle očekávání. Po zapsání do kalendáře se uloží. Uživatel si může zadat tělesné parametry a aplikace si ho ukládá po registraci.

Co ovšem nemohlo být realizováno, byla zpětná vazba aplikace uživateli. Na počátku vývoje této idey nastávalo nespočet problémů, jež poté znemožnili chod celé aplikace. Je to taky hlavní kritický problém, který se mi při vývoji nepodařil vyřešit. Jednou z nápadů na využití úpravy tělesných parametrů byly grafy, které na počátku byly velikou obtíží k opravě a do výsledné aplikace se nedostaly.

Aplikace by se určitě dala vylepšit, ať už zmíněnou komunikací s uživatelem, hezčím vzhledem, vylepšeným kalendářem anebo přidáním dalších typů tréninků. Veliké ponaučení, které jsem si ze psaní ročníkového projektu odnesl je, že nejtěžším částem projektu je nejlepší čelit co nejdřív. Každopádně mi tvorba aplikace přinesla mnoho znalostí a jsem rád, že jsem se do projektu pustil.

Zdroje

- [1] Wikimedia Foundation. (2024, October 25). *Visual Studio Code*. Wikipedia. https://cs.wikipedia.org/wiki/Visual_Studio_Code [cit. 2025-03-21]
- [2] Wikimedia Foundation. (2025, March 15). *Python*. Wikipedia. <https://cs.wikipedia.org/wiki/Python> [cit. 2025-03-21]
- [3] Django Software Foundation. (2005). *Django*. Django Project. <https://www.djangoproject.com/> [cit. 2025-03-22]
- [4] Django Software Foundation. (2005). *Templates: Django documentation*. Django Project. <https://docs.djangoproject.com/en/5.1/topics/templates/#:~:text=A%20Django%20template%20is%20a,is%20rendered%20with%20a%20context> [cit. 2025-03-22]
- [5] W3Schools. (1998). *W3schools.com*. W3Schools Online Web Tutorials. <https://www.w3schools.com/bootstrap5/#:~:text=Bootstrap%20is%20the%20newest,Start%20learning%20Bootstrap%20now%20C2%BB> [cit. 2025-03-23]
- [6] Christie, T. (2024, March 22). *Django rest framework*. Home - Django REST framework. <https://www.django-rest-framework.org/> [cit. 2025-03-23]
- [7] Wikimedia Foundation. (2025, March 26). *Frontend and backend*. Wikipedia. https://en.wikipedia.org/wiki/Frontend_and_backend [cit. 2025-03-26]
- [8] GitHub. (2024, October 21). *Gyarab/2024-4E-Pesula-TreninkDenik*. gyarab/2024-4e-pesula-TreninkDenik. <https://github.com/gyarab/2024-4e-pesula-TreninkDenik> [cit. 2025-03-27]
- [9] Christie, T. (2016, January 27). *Third Party packages*. Third Party Packages - Django REST framework. <https://www.django-rest-framework.org/community/third-party-packages/#serializers> [cit. 2025-03-28]
- [10] Dostalová, Z. (2014, July 1). *Frontend vs. backend*. Czechitas. <https://www.czechitas.cz/blog/frontend-vs-backend> [cit. 2025-03-28]

[11] GeeksforGeeks. (2023, November 17). *Python: Calendar module*. GeekforGeeks <https://www.geeksforgeeks.org/python-calendar-module/> [cit. 2025-03-28]

[12] w3resource. (2023). *Python TextCalendar Module: monthcalendar() method*. Python calendar module: Monthcalendar() method. <https://www.w3resource.com/python/module/calendar/monthcalendar.php> [cit. 2025-03-28]

[13] ivory54321 (Ed). (2022, July 22). *R/django on reddit: Unable to save a ModelForm: IntegrityError: UNIQUE constraint failed: home_profile.user_id*. Reddit. <https://www.reddit.com/r/django/comments/w584vd/comment/ih8n8lc/?rdt=45656> [cit. 2025-03-29]

[14] Affectionate-Ad-7865 (Ed.). (2022, December 18). *R/django on reddit: Href="{%url "something" %}" doesn't work*. Reddit. https://www.reddit.com/r/django/comments/zp36ku/hrefurl_something_doesnt_work/ [cit. 2025-03-29]

Seznam obrázků

Obr. 1. – logo Visual Studio Code

(zdroj: Wikimedia Foundation. (n.d.). *File:visual studio code 1.35 icon.svg*. Wikipedia. https://en.m.wikipedia.org/wiki/File:Visual_Studio_Code_1.35_icon.svg [cit. 2025-03-22])

Obr. 2. – logo frameworku Django

(zdroj: Titov, A. (2025, February 17). *Django logo*. 1000 Logos The Famous logos and Popular company logos in the World Django Logo Comments. <https://1000logos.net/django-logo/> [cit. 2025-03-23])

Obr. 3. & 4. – kódy formulářů v aplikaci

(zdroj: GitHub. (2025, March 8). *Gyarab/2024-4E-Pesula-TreninkDenik*. gyarab/2024-4e-pesula-TreninkDenik. <https://github.com/gyarab/2024-4e-pesula-TreninkDenik/blob/main/treninkdenik/api/forms.py> [cit. 2025-03-26])

Obr. 5. & 6. – kód tvorby kalendáře a funkce zapistreninku

(zdroj: GitHub. (2025, March 8). *Gyarab/2024-4E-Pesula-TreninkDenik*. gyarab/2024-4e-pesula-TreninkDenik. <https://github.com/gyarab/2024-4e-pesula-TreninkDenik/blob/main/treninkdenik/api/views.py> [cit. 2025-03-27])

Obr. 7. – vzhled kalendáře

Obr. 8. – struktura seznamu tréninků

Obr. 9. – seřazený seznam příkazů pro instalaci aplikace

(zdroj: GitHub. (2024, October 21). *Gyarab/2024-4E-Pesula-TreninkDenik*. gyarab/2024-4e-pesula-TreninkDenik. <https://github.com/gyarab/2024-4e-pesula-TreninkDenik/blob/main/README.md> [cit. 2025-03-28])