

ZÁVĚREČNÁ STUDI JNÍ PRÁCE

dokumentace



FLYTE - fitness webová aplikace

Autor: Viktorie Mazurová
Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování
Třída: IT4
Školní rok: 2024/25

Poděkování

Ráda bych poděkovala fitness Contours, ve kterém jsem pracovala, za nápad na funkčnost aplikace. Obzvláště děkuji své mamce Markétě Mazurové, která mi dala celou vizi na to Flyte zrealizovat.

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl 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 14. 1. 2025

.....

Podpis autora

Abstrakt

Flyte je webová aplikace, která umožňuje uživatelům přihlásit se do systému, registrovat se na kurzy, sledovat svůj tréninkový plán a procházet personalizované jídelníčky. Díky těmto funkcím aplikace usnadňuje nejen práci trenérů, ale také poskytuje klientům lepší přehled o jejich aktivitách a pokrocích.

Aplikace je navržena tak, aby byla uživatelsky přívětivá. Dokumentace detailně popisuje jednotlivé kroky vývoje aplikace – od návrhu databázového modelu, přes implementaci funkcí pro registraci a správu kurzů, až po tvorbu nástrojů pro sledování tréninků a jídelníčků.

Flyte je výsledkem mé snahy vytvořit systém, který by zlepšil každodenní provoz fitness centra a spokojenost jak jeho klientů, tak i zaměstnanců.

Abstract

During my work at a fitness center, I realized that the process of managing courses, training sessions, and client information often lacked efficiency and a modern approach. This motivated me to create Flyte – a system designed to help fitness centers improve organization while offering users a simple way to manage their training and nutrition plans.

Flyte is a web application that allows users to log into the system, register for courses, track their training schedules, and browse personalized meal plans. With these features, the application not only streamlines trainers' workflows but also provides clients with a better overview of their activities and progress.

The application is designed to be user-friendly and responsive, ensuring it can be used on various devices. The documentation provides a detailed overview of the development process – from designing the database model and implementing features for course registration and management to creating tools for tracking training sessions and meal plans.

Flyte is the result of my effort to develop a system that enhances the day-to-day operations of fitness centers and improves the satisfaction of both clients and staff.

Obsah

Úvod	6
1 Teoretická a metodická východiska	7
1.1 Aplikace pro zdraví a fitness	7
1.2 Architektura aplikace	7
1.3 Počáteční zkušenosti	7
2 Využité technologie	8
2.1 Lucidchart	8
2.2 SQLite	8
2.3 Strapi API	8
2.4 React	9
2.5 Tailwind CSS	9
2.6 FullCalendar	9
2.7 ApexCharts	9
3 Způsoby řešení na straně backendu	10
3.1 Návrh databáze	10
3.2 Implementace backendu pomocí Strapi	11
4 Způsoby řešení na straně frontednu	12
4.1 Vývoj frontendu pomocí React.js	12
4.2 Propojení databáze s frontendem	13
4.3 Zobrazení kurzů pomocí metody GET	13
4.4 Tréninkové plány	15
4.5 Jídelníčky	16
4.6 Vykreslení grafů	19

5	VÝSLEDKY ŘEŠENÍ, VÝSTUPY, UŽIVATELSKÝ MANUÁL	20
5.1	Funkce aplikace	20
5.2	Splnění a nesplnění cíle	21

ÚVOD

Když jsem pracovala ve fitness centru, často jsem si říkala, že by to všechno mohlo být jednodušší. Přihlašování na kurzy, plánování tréninků nebo sledování jídelníčků bylo zbytečně složité a zabíralo hodně času. Klienti i trenéři se často ztráceli ve starých, nepřehledných systémech a to mě motivovalo vytvořit něco lepšího – webovou aplikaci, která by všechny tyhle problémy vyřešila. Proto jsem si tohle téma vybrala jako svoji maturitní práci.

Mým cílem bylo vytvořit aplikaci, kde se klienti můžou snadno přihlásit na kurzy přes jednoduchý kalendář, sledovat své tréninkové plány a mít přehled o svých jídelníčcích. Trenéři by si zase mohli jednoduše spravovat kurzy a vytvářet stravovací plány pro své klienty. Chtěla jsem, aby to bylo přehledné, rychlé a mohlo to fungovat na všech zařízeních.

Na vývoj jsem použila nástroje, které jsem dříve vůbec neznala, ale práce na této aplikaci mě donutila se je naučit. Frontend jsem postavila na React.js, což mi umožnilo vytvořit jednoduché a hezky vypadající uživatelské prostředí. Backend jsem řešila přes Strapi API, kde se ukládají všechna data. K tomu jsem použila Tailwind CSS, aby aplikace vypadala moderně a dala se snadno přizpůsobit.

V dokumentaci popisuju, jak jsem postupovala od návrhu aplikace, přes vývoj databáze, až po funkce jako přihlašování, rezervace kurzů a sledování jídelníčků. Na konci hodnotím výsledky a ukazuju, jak aplikace funguje.

Tahle aplikace je podle mě dobrým příkladem, jak moderní technologie můžou zlepšit fungování fitness centra a zároveň klientům usnadnit cestu ke zdravému životnímu stylu. Jsem ráda, že jsem mohla něco takového vytvořit, a doufám, že to někomu usnadní práci.

1 TEORETICKÁ A METODICKÁ VÝCHODISKA

1.1 APLIKACE PRO ZDRAVÍ A FITNESS

V dnešní době jsou aplikace zaměřené na zdraví a fitness stále populárnější. Umožňují lidem sledovat svůj pokrok, plánovat tréninky a upravovat jídelníčky. Přestože podobných aplikací existuje spousta, často jim chybí přehlednost nebo možnost přizpůsobit je konkrétním potřebám uživatelů. Proto jsem se rozhodla vytvořit aplikaci, která by nabídla snadné přihlašování na kurzy, správu tréninků a sledování jídelníčků, a to vše v prostředí, které je jednoduché a příjemné na ovládání.

1.2 ARCHITEKTURA APLIKACE

Aplikace je navržena tak, aby frontend a backend byly oddělené. Tento přístup umožňuje lepší správu dat, jednodušší údržbu a možnost aplikaci dále rozšiřovat. Díky tomu můžu jednotlivé části vyvíjet a aktualizovat nezávisle na sobě, což nejen zjednodušuje práci, ale také urychluje přidávání nových funkcí. Tento způsob mi připadal nejvhodnější, protože zajišťuje, že aplikace bude flexibilní a připravená na budoucí rozvoj.

1.3 POČÁTEČNÍ ZKUŠENOSTI

S vývojem aplikací jsem se už dříve setkala, například při práci na projektu v Django, kde jsem vytvářela jednoduchý uživatelský systém. Přesto byl tento projekt pro mě velkou výzvou, protože technologie jako React a Strapi jsem předtím nikdy nepoužila. Musela jsem se je naučit od základů a postupně zjistit, jak je správně využít pro potřeby mé aplikace. I když to bylo náročné, získané zkušenosti mi hodně pomohly a ukázaly mi nové možnosti, jak přistupovat k vývoji moderních webových aplikací.

2 VYUŽITÉ TECHNOLOGIE

Při řešení tohoto projektu byla použita kombinace technologií, které mi umožnily snadný vývoj a implementaci webové aplikace pro fitness centrum. Výběr každé technologie byl vždy velmi zvážěn s ohledem na její funkcionalitu, dostupnost a schopnost splnit požadavky projektu.

2.1 LUCIDCHART

Lucidchart je nástroj pro vizuální návrh diagramů a schémat. V rámci tohoto projektu byl použit pro návrh databázového modelu, což byla základní a jedna z nejhlavnějších částí při vývoji aplikace. Jeho snadné a efektivní uživatelské rozhraní umožnilo snadno vytvářet přehledný návrh databázových tabulek a jejich relací.

2.2 SQLITE

SQLite - lehká a samostatná relační databáze. V mém projektu slouží pro ukládání dat o uživatelích, kurzech, tréninkových plánech a jídelních doporučeních. SQLite nevyžaduje server, což zjednodušuje její nasazení. Vybrala jsem SQLite kvůli její jednoduchosti a podpoře Strapi API. Jiné databáze se mi zdály pro můj projekt moc zbytečně složité.

2.3 STRAPI API

Strapi je open-source headless CMS (Content Management System), které poskytuje flexibilní backend pro správu dat a API. V tomto projektu je Strapi využito jako backendový systém pro správu celé aplikace - kurzů, tréninkových plánů, jídelníčků a uživatelských dat. Strapi jsem zvolila díky jeho schopnosti rychle vytvářet a spravovat API, což mi usnadnilo komunikaci mezi frontendem a databází pomocí Rest API fetch dotazů.

2.4 REACT

React.js je JavaScriptová knihovna určená pro tvorbu uživatelských rozhraní. V tomto projektu jsem React využila k vývoji frontendové části aplikace, hlavně díky jeho flexibilnímu přístupu, snadné práci s komponentami, velké dokumentaci a interaktivitu. Přemýšlela jsem o využití Next.js místo Reactu, ale protože jsem se především chtěla naučit React, zvolila jsem jej.

2.5 TAILWIND CSS

Tailwind CSS je utility-first CSS framework, který umožňuje rychlé a flexibilní stylování uživatelského rozhraní. V mém projektu byl použit pro stylování aplikace, což umožnilo dosáhnout moderního a responzivního designu. Tailwind jsem vybrala kvůli jeho efektivitě při psaní stylů, schopnosti rychlého prototypování a snadnému přizpůsobení designu podle specifických potřeb projektu.

2.6 FULLCALENDAR

FullCalendar je JavaScriptová knihovna určená k vytváření interaktivních kalendářů. V mé aplikaci je využita k zobrazení a správě přihlašování na fitness kurzy, umožňující uživatelům snadno zobrazit dostupné termíny a rezervovat si místo. FullCalendar jsem zvolila pro jeho bohatou funkcionalitu, vysokou přizpůsobitelnost a snadnou integraci do React aplikace.

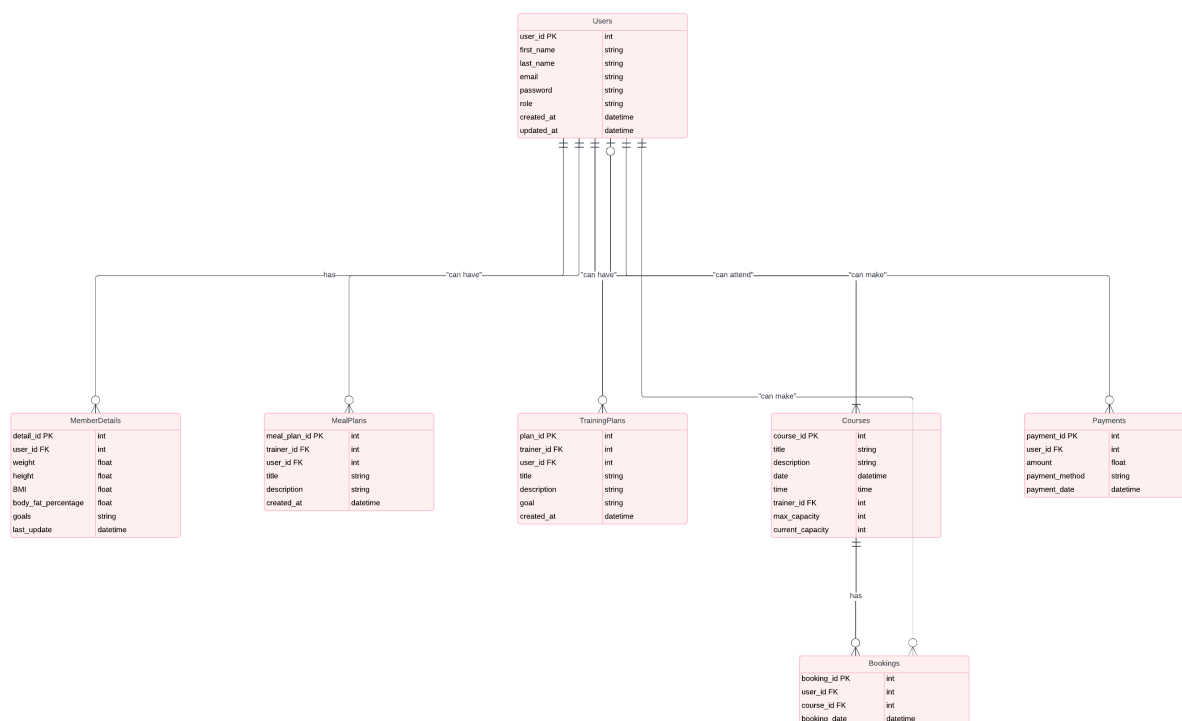
2.7 APEXCHARTS

ApexCharts je knihovna pro tvorbu interaktivních grafů a vizualizaci dat. V mém projektu slouží k zobrazení zdravotních metrik, jako je BMI a procento tělesného tuku, v přehledné grafické podobě. ApexCharts jsem vybrala díky jeho schopnosti vytvářet moderní, responzivní a snadno konfigurovatelné grafy, které umožňují uživatelům lépe porozumět a interpretovat jejich data.

3 ZPŮSOBY ŘEŠENÍ NA STRANĚ BACKENDU

3.1 NÁVRH DATABÁZE

Prvním krokem bylo vytvoření databázového modelu, který bude základem celé aplikace. Pomocí Lucidchartu jsem si navrhla všechny tabulky, jejich atributy a vztahy mezi nimi. Návrh zahrnoval klíčové entity, jako jsou: uživatelé, což jsou klienti a trenéři, přičemž každý z nich má přiřazenou specifickou roli a oprávnění. Dále jsou zde kurzy, které obsahují informace jako název, čas konání a Tréninky slouží k uchování historie a plánů cvičení, zatímco jídelníčky představují personalizované stravovací plány pro jednotlivé klienty. Detail uživatele umožňuje sledování pokroků v oblasti zdraví a fitness. Kromě toho databáze zahrnuje také platby, které mají zaznamenávat provedené transakce, ale toto zatím není v aplikaci zprovozněno.



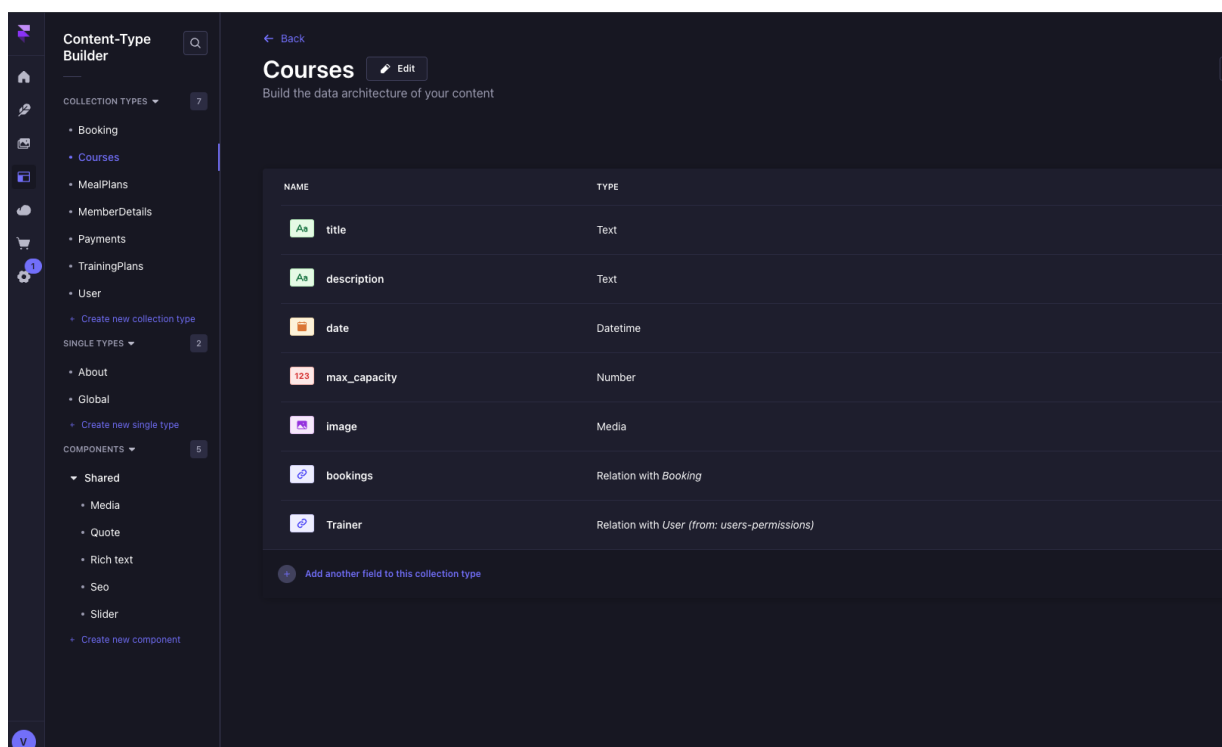
Obrázek 3.1.1: Databázové schéma

3.2 IMPLEMENTACE BACKENDU POMOCÍ STRAPI

Strapi jsem zvolila jako backendový systém především kvůli jeho uživatelsky přívětivému rozhraní. Tato flexibilita a jednoduchost jsou ideální pro projekty, kde je potřeba rychle vytvořit administrativní rozhraní a mít plnou kontrolu nad databází. Po nastavení základního prostředí a instalaci Strapi jsem začala vytvářet kolekce, což jsou základní bloky pro ukládání dat v aplikaci. Například v aplikaci pro správu tréninků jsem vytvořila kolekci pro uživatele, kolekci pro tréninky a také samozřejmě kurzy a jídelníčky.

Když byly kolekce vytvořeny, začala jsem přidávat jednotlivá pole pro každou kolekci, aby byla data co nejvíce strukturovaná a odpovídala požadavkům aplikace. Pole jsem vybírala různých typů – textová, číselná, datová, vztahová (pro propojení s jinými kolekcemi) a další. Tento proces je důležitý, protože zajišťuje, že aplikace bude schopná správně zpracovávat a zobrazovat data. Pole jsou nastavena tak, aby uživatelé mohli snadno přidávat nové informace nebo upravovat existující, a zároveň aby byla zajištěna validita dat.

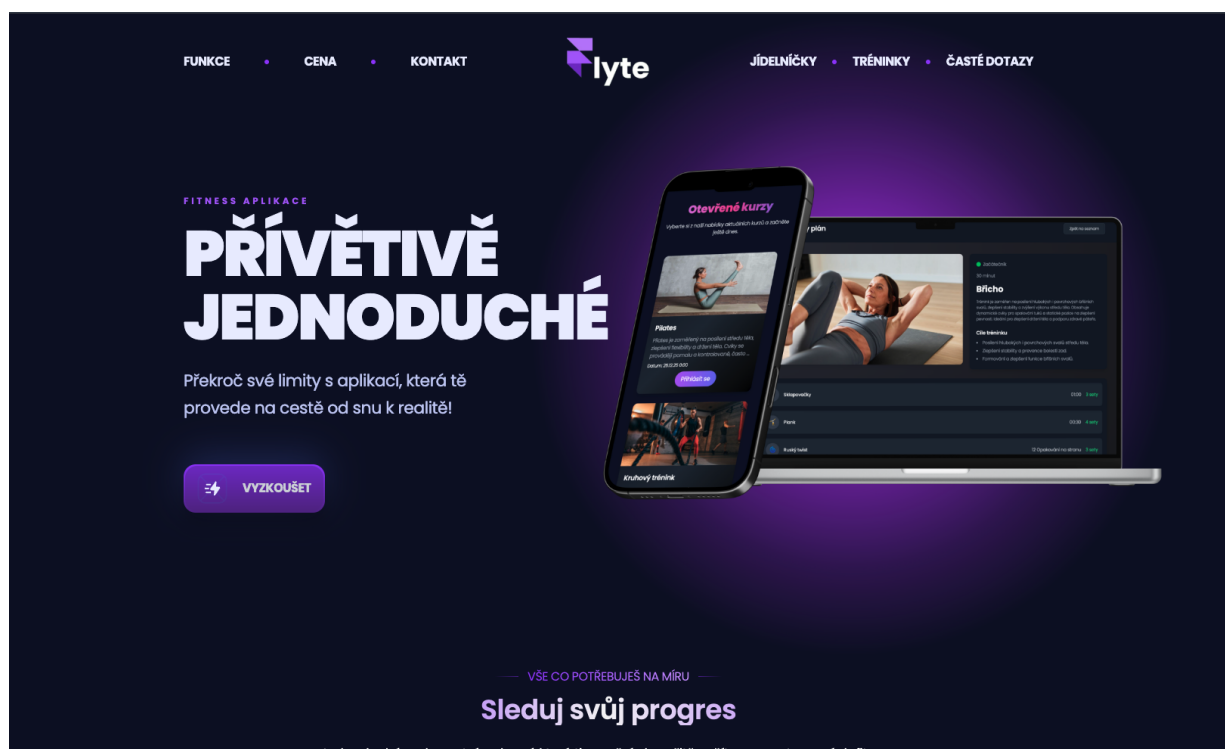
Další významnou součástí Strapi je správa uživatelských rolí a práv. Role umožňují různým uživatelům přístup pouze k těm částem aplikace, které mají povoleny. Pro můj projekt jsem definovala role pro trenéry, klienty a jednoho hlavního trenéra - správce daného fitka. A každé roli jsem přiřadila specifická oprávnění. Například trenéři mají přístup k vytváření a úpravám tréninkových plánů. Takto jsem mohla přesně určit, co jednotliví uživatelé mohou a nemohou vidět, což zvyšuje bezpečnost a soukromí dat v aplikaci.



4 ZPŮSOBY ŘEŠENÍ NA STRANĚ FRONTEDNU

4.1 VÝVOJ FRONTENDU POMOCÍ REACT.JS

S funkčním backendem jsem začala pracovat na frontendu. Jako první jsem vytvořila úvodní stránku. Bylo to záměrné rozhodnutí, protože jsem s Reactem předtím neměla žádné zkušenosti, a chtěla jsem si nejdříve osvojit jeho základy. Úvodní stránku jsem rozdělila do několika sekcí, což mi pomohlo pochopit, jak se pracuje s komponentami a jak jednotlivé části aplikace propojit. Postupně jsem přidávala navigaci, texty a obrázky, abych si vyzkoušela, jak vše stylovat a strukturovat.



Obrázek 4.1.1: Hlavní stránka

4.2 PROPOJENÍ DATABÁZE S FRONTENDEM

Dalším krokem bylo vytvoření funkcí pro registraci a přihlášení uživatelů. Propojila jsem aplikaci s backendem pomocí fetch požadavků a zajistila, aby se data uživatelů ukládala a ověřovala v databázi. Implementovala jsem také validace vstupních dat, aby aplikace zvládla zpracovat pouze správné informace a také aby bylo zajištěno, že heslo uživatelů splňuje požadavky. Toto jsem zpracovala pomocí RegEx požadavků. Po přihlášení se uživatelům zobrazují funkce přizpůsobené jejich roli – klientům jejich tréninky a kurzy, trenérům možnosti přidání kurzu, odstranění kurzu, přidání tréninku atp.

4.3 ZOBRAZENÍ KURZŮ POMOCÍ METODY GET

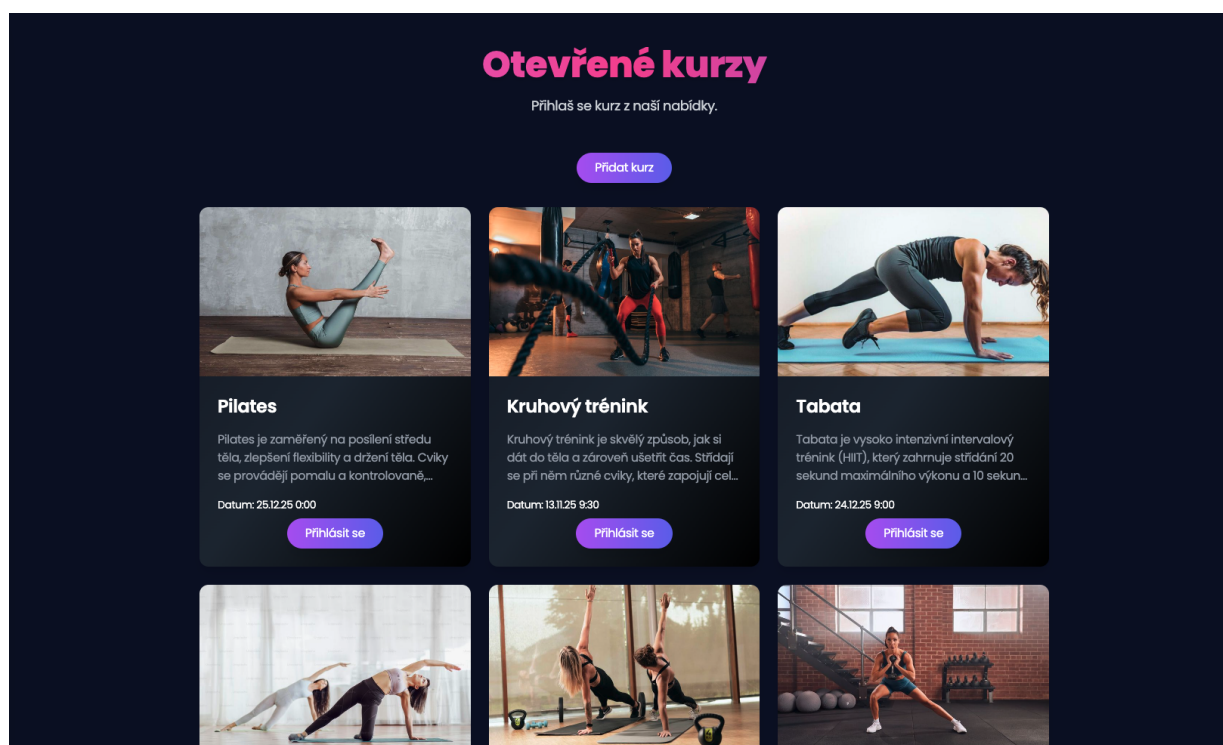
Po vytvoření základní struktury aplikace jsem se zaměřila na vytvoření stránky pro zobrazení všech kurzů, které ještě neproběhly. Tento krok byl klíčový, protože uživatelé, konkrétně trenéři a klienti, potřebují mít přehled o dostupných kurzech, které si mohou rezervovat. Stránka je dynamická a automaticky zobrazuje pouze kurzy, které ještě nejsou datově uzavřeny, tedy které nejsou po aktuálním systémovém datu. Kurzy, které již probíhají nebo jsou po datu konání, jsou skryté pro všechny uživatele.

Pro zajištění interaktivity mezi aplikací a databází jsem použila REST API, což je rozhraní pro komunikaci mezi klientskou aplikací a serverem. Pomocí REST API jsem mohla načítat data o jednotlivých kurzech z databáze a následně je dynamicky zobrazovat na stránce.

Každá karta kurzu navíc obsahuje odkaz, který vede na detailní stránku kurzu. Tato stránka poskytuje podrobnější informace, jako jsou přesný čas, datum a počet volných míst.

Pro roli trenéra jsem přidala speciální funkce na stránce detailu kurzu. Trenér má přístup k tabulce s přihlášenými uživateli, kde se zobrazují jména všech osob, které jsou na daný kurz přihlášeny. Pokud je potřeba, trenér má možnost odstranit uživatele z kurzu, například pokud se někdo neúčastní nebo se na kurz přihlásil omylem.

Dále jsem do detailní stránky kurzu zahrnula možnost odstranění kurzu. Tuto možnost mají pouze trenéři, protože administrují kurzy a mohou se rozhodnout zrušit konkrétní kurz, pokud například nastanou nepředvídané okolnosti, jako změna místa nebo času, nebo pokud není dostatek účastníků.



Obrázek 4.3.1: Zobrazení všech kurzů

4.3.1 Vytvoření kurzů pomocí metody POST

Pro trenéry jsem implementovala funkci přidávání nových kurzů, která jim umožňuje jednoduše zadat potřebné údaje o kurzu. Vytvořila jsem formulář, který kontroluje, že všechny povinné údaje, jako je název, datum, čas a místo, jsou správně vyplněné. Pokud některý z těchto údajů chybí nebo je nesprávně vyplněn, formulář trenérovi zobrazí chybu a neumožní odeslání formuláře, dokud nebudou všechny údaje opraveny.

Po úspěšném odeslání formuláře jsou data o novém kurzu uložena do databáze a kurz se okamžitě zobrazuje v seznamu dostupných kurzů. Tento proces je automatizovaný a trenéři mají okamžitou zpětnou vazbu, že kurz byl úspěšně přidán pomocí ToastContainer.

4.3.2 Vytvoření kalendáře pro zobrazení kurzů

jedním z posledních kroků, co se týkalo kurzů bylo vytvoření kalendáře, který přehledně zobrazuje všechny dostupné kurzy podle data. Původně jsem zvažovala vytvoření vlastního kalendáře, ale po zhodnocení jsem se rozhodla využít knihovnu FullCalendar, která nabízí hotové řešení, které je snadno přizpůsobitelné. Kalendář jsem propojila s backendem ve Strapi pomocí API, které poskytuje data o kurzech.



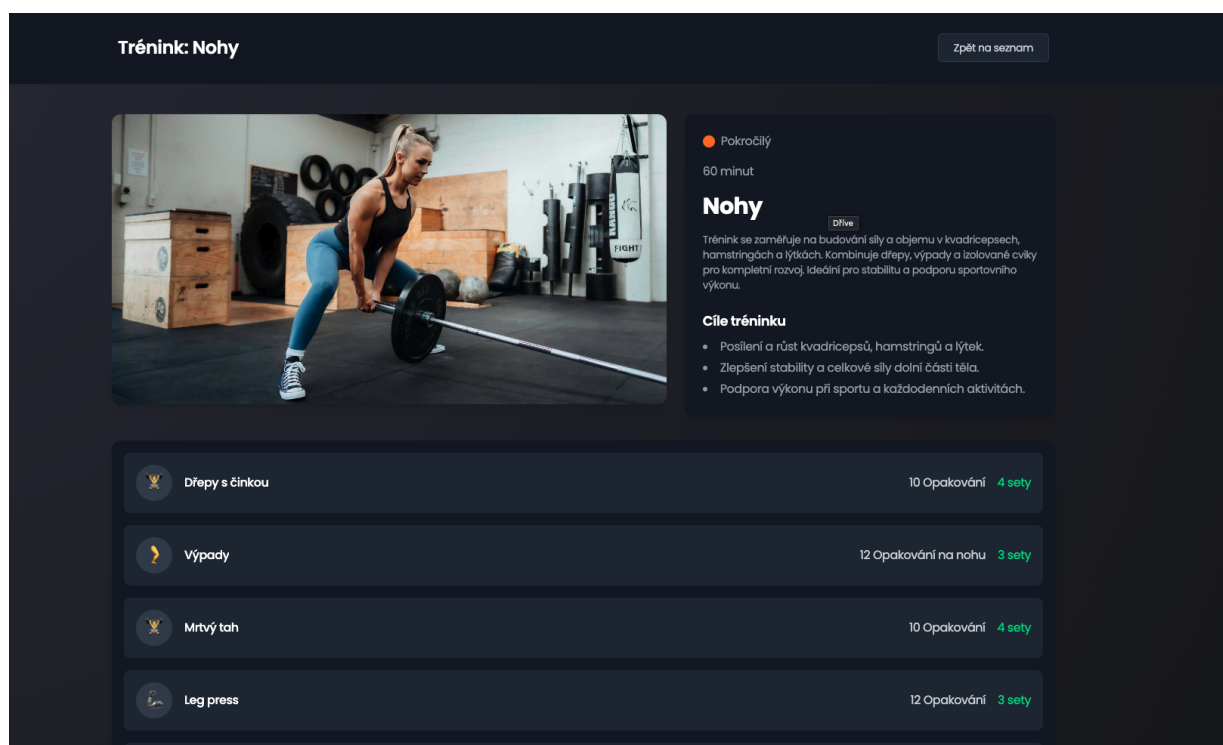
Obrázek 4.3.2: Kalendář kurzů

4.4 TRÉNINKOVÉ PLÁNY

Práce na tréninkových plánech byla dalším důležitým krokem při vývoji aplikace. Nejdříve jsem vytvořila stránku, kde se zobrazuje seznam všech tréninkových plánů. Chtěla jsem, aby design vypadal jinak než u kurzů, proto jsem ho zvolila více minimalistický. U každého plánu jsem zobrazila jen základní informace, jako je název, obtížnost, čas a malý popis s možností kliknout na detail.

4.4.1 Detail tréninku pomocí metody GET

Detailní stránka tréninkového plánu ukazuje uživateli všechny cviky včetně jejich popisu a opakování. Tyto informace jsou rozděleny přímo na frontendu, takže uživatel vidí vše přehledně uspořádané. Aby ale byla databáze jednodušší a nebylo nutné vytvářet složité struktury, rozhodla jsem se ukládat jednotlivé cviky v plánech jako jeden string. V tomto stringu jsou jednotlivé části plánu oddělené středníkem ; a různé atributy, jako název cviku nebo počet opakování, oddělené rourou |. Na frontendu jsem potom tento string rozdělila a zobrazila uživatelům přehledný seznam tréninků.



Obrázek 4.4.1: Zobrazení detailu tréninkových plánů

4.4.2 Vytvoření tréninkového plánu pomocí metody POST

Pro trenéry jsem přidala možnost vytvářet nové tréninkové plány. Formulář umožňuje zadávat cviky včetně jejich atributů a ukládá je do databáze jako string se stejnou strukturou. Kód pro zpracování těchto dat jsem našla na internetu, protože takové zpracování stringů bylo pro mě dost složité. Díky tomu jsem ale mohla rychle implementovat funkci, která funguje efektivně a jednoduše.

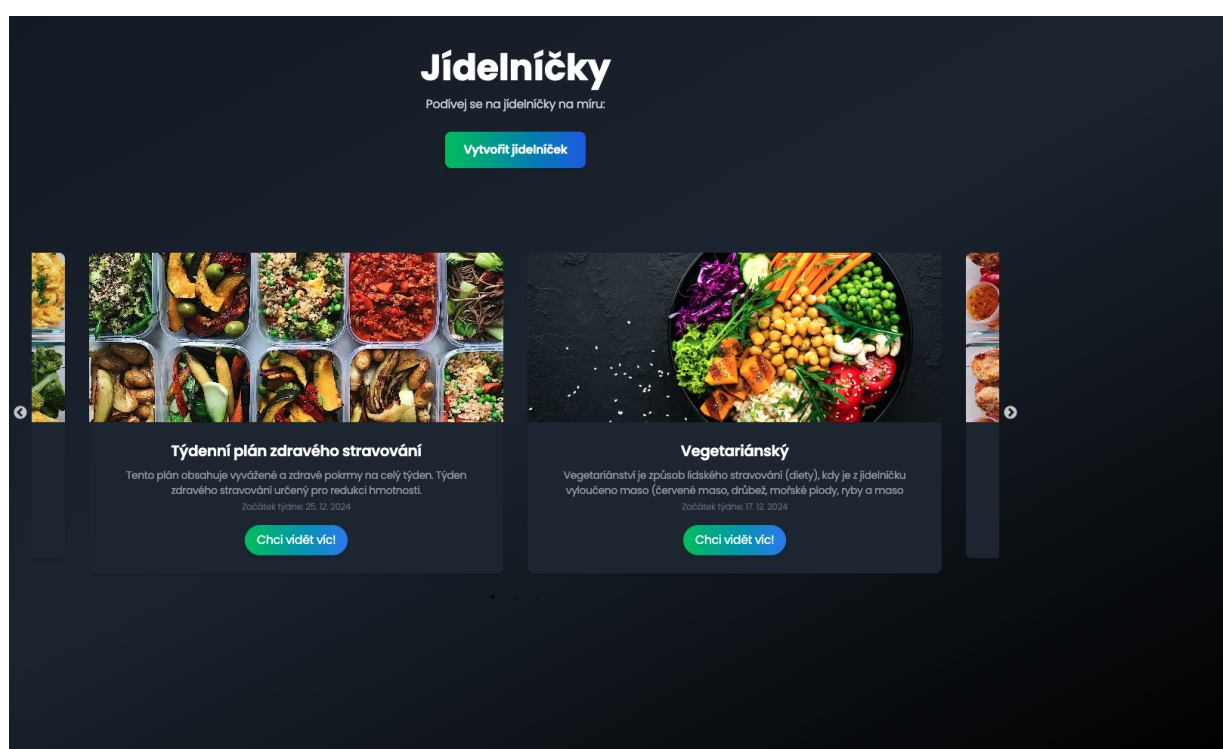
4.5 JÍDELNÍČKY

Práce na jídelníčcích byla velmi podobná tomu, co jsem vytvořila u tréninkových plánů, ale s jiným zobrazením a přístupem k datům. Tentokrát jsem se rozhodla ukládat data v oddělených polích, což mělo usnadnit jejich zpracování na backendu i frontendu. Struktura databáze byla díky tomu přehlednější a lépe přizpůsobená pro práci s jednotlivými položkami jídelníčků. Nicméně tento přístup se ukázal jako časově náročnější, než jsem očekávala.

Zpětně bych zvolila stejný způsob ukládání dat jako u tréninkových plánů, tedy ve formě jednoho stringu, který by se následně rozdělil na frontendu. Tento přístup by byl jednodušší na implementaci a ušetřil by mi spoustu času při vývoji.

4.5.1 Zobrazení všech jídelníčků pomocí metody GET

Na frontendu jsem opět začala vytvořením seznamu jídelníčků, kde se zobrazují základní informace, jako je název a datum vytvoření. I tady jsem zvolila jiný design a to pomocí slideru. Každý jídelníček má také svou detailní stránku, kde uživatelé vidí jednotlivá jídla.



Obrázek 4.5.1: Zobrazení jídelníčků

4.5.2 Vytvoření jídelníčku pomocí metody POST

Pro trenéry jsem vytvořila funkci na přidávání nových jídelníčků. Na rozdíl od tréninkových plánů, kde jsem používala stringy, se u jídelníčků data ukládají rovnou do oddělených polí. Formulář umožňuje zadávat jednotlivá jídla, jejich popis a doporučení pro klienty. Po odeslání formuláře se data ukládají do databáze pomocí metody POST.

Vytvořte jídelníček pro své klienty pomocí tohoto formuláře.
Vyplňte název jídelníčku, vyberte den a přidejte jídla pro jednotlivé části dne.
Každé jídlo může mít zadanou gramáž a přílohu.

Název jídelníčku

Obrázek jídelníčku

Vybrat soubor Soubor nevybrán

Pondělí

Snídaně Zadejte jídlo Gramy Přidat další	Dopolední svačina Zadejte jídlo Gramy Přidat další	Oběd Zadejte jídlo Gramy Přidat další	Odpolední svačina Přidat další	Večeře Přidat další
--	--	---	--	-------------------------------

Úterý

Snídaně Přidat další	Dopolední svačina Přidat další	Oběd Přidat další	Odpolední svačina Přidat další	Večeře Přidat další
--------------------------------	--	-----------------------------	--	-------------------------------

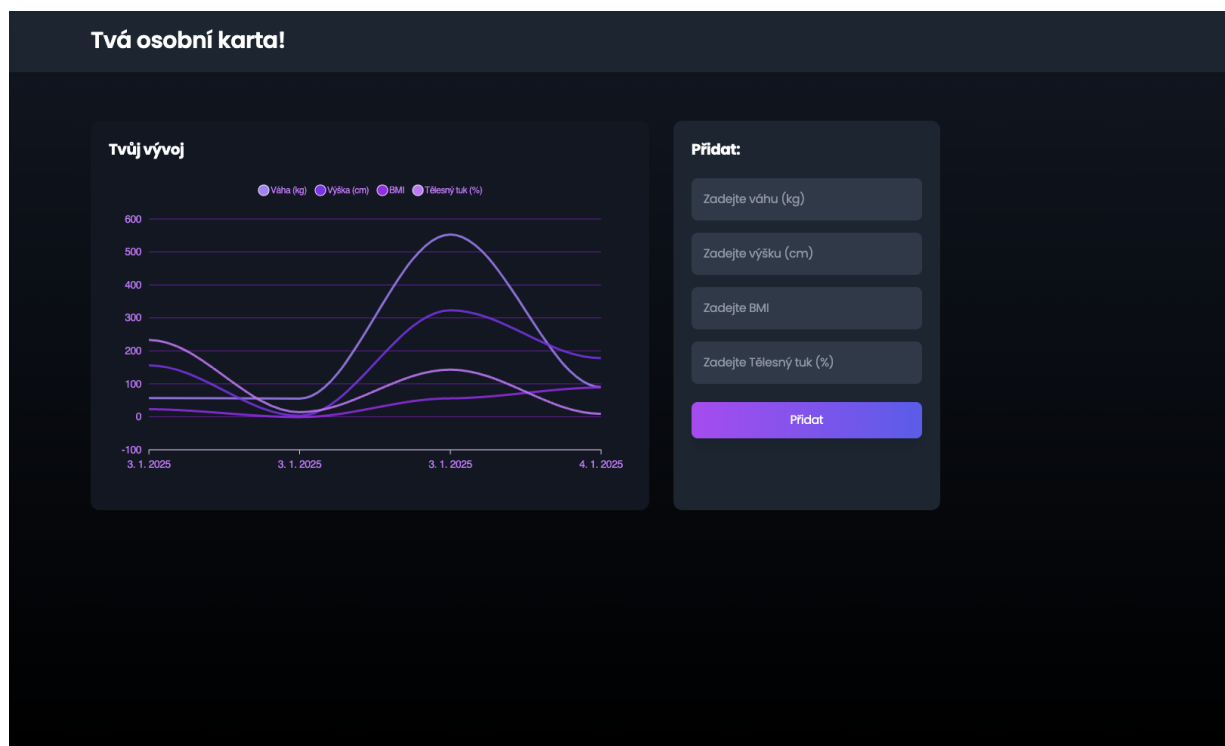
Obrázek 4.5.2: Přidání nového jídelníčku

4.6 VYKRESLENÍ GRAFŮ

Na závěr jsem se pustila do práce na zobrazení detailů uživatele, kde jsem chtěla zobrazit základní údaje, jako jsou výška, váha, BMI a tělesný tuk. Rozhodla jsem se tyto informace vizualizovat pomocí grafů, protože jsem si chtěla vyzkoušet práci s Apex Charts.

Data jsou v databázi ukládána přímo k jednotlivým uživatelům, což zajišťuje, že každý má své hodnoty. Na frontendu jsem pak pomocí Apex Charts vykreslovala náhled, kde se zobrazovaly tyto parametry v čase.

Toto jsem bohužel zvládla jen základně, protože už jsem měla málo času a práce s grafy nebyla úplně jednoduchá. Myslím si ale, že tato část by mohla být v budoucnu dále rozvinuta, aby aplikace nabízela uživatelům ještě lepší přehled.



Obrázek 4.6.1: Přidání nového jídelníčku

5 VÝSLEDKY ŘEŠENÍ, VÝSTUPY, UŽIVATELSKÝ MANUÁL

5.1 FUNKCE APLIKACE

Při příchodu na úvodní stránku aplikace Flyte si uživatel může vybrat mezi registrací a přihlášením. Po přihlášení získá přístup ke všem funkcím, které se liší podle jeho role – klienta nebo trenéra.

Stránka kurzů zobrazuje přehled dostupných lekcí s detaily, jako je název, čas. Uživatel se může na kurz přihlásit, zatímco trenér má možnost přidávat nové lekce. Kalendář zobrazuje všechny kurzy přehledně podle data a umožňuje uživatelům kliknout na detail nebo se přihlásit.

Treninkové plány vytvářejí trenéři, podle kterých se mohou uživatelé orientovat během cvičení.

Na stránce jídelníčků klient vidí své stravovací plány rozdělené podle částí dne, jako je snídane nebo večeře. Trenéři mohou přidávat nové jídelníčky.

Osobní složka poskytuje možnost pozorovat svůj progres pomocí údajů, jako jsou: váha, výška, tělesný tuk a BMI, přičemž každý má přístup pouze ke svým datům.

V aplikaci Flyte byl kladen důraz na atraktivní rozhraní, aby byla aplikace snadno použitelná jak pro klienty, tak pro trenéry. Flyte nabízí komplexní základ pro správu fitness center s potenciálem dalšího rozšíření a vylepšení.

5.2 SPLNĚNÉ A NESPLNĚNÉ CÍLE

Cílem bylo vytvořit aplikaci Flyte, která by zjednodušila správu fitness aktivit a umožnila klientům i trenérům efektivně pracovat s kurzy, tréninkovými plány a jídelníčky a monitorovat své pokroky. Většinu těchto cílů se mi podařilo splnit, a aplikace v současnosti funguje podle očekávání. Některé části však stále nejsou dokončené a aplikace by mohla být do budoucna ještě přehlednější a intuitivnější pro uživatele.

Do budoucna bych se ráda zaměřila na rozšíření aplikace o další funkce. Mezi potenciální vylepšení patří přidání možnosti nastavení uživatelského účtu, implementace přihlašování pomocí OAuth, a vylepšení grafických přehledů o aktivitách, například detailnějšími grafy pro sledování pokroku a také zprovoznění plateb.

Na závěr jsem se také pustila do nastavení aplikace, které by uživatelům umožnilo například změnit své jméno. Tato část zatím funguje pouze na frontendu, protože se mi ji nepodařilo plně propojit s backendem. Původně mělo toto nastavení zahrnovat možnost změny křestního jména, příjmení, e-mailu a také hesla. Tyto funkce aplikace nabízí, ale bohužel zatím nejsou plně funkční. Do budoucna se zaměřím na jejich dokončení a propojení s databází, což umožní uživatelům plně spravovat své údaje.

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- [1] *React Documentation* [online] Odkaz: <https://reactjs.org/docs/>
- [2] *Strapi Documentation* [online] Odkaz: <https://docs.strapi.io/>
- [3] *Tailwind CSS Documentation* [online] Odkaz: <https://tailwindcss.com/docs/>
- [4] *FullCalendar Documentation* [online] Odkaz: <https://fullcalendar.io/docs/>
- [5] *Apex Charts Documentation* [online] Odkaz: <https://apexcharts.com/docs/>
- [6] The Net Ninja. *React Tutorials* [online] Odkaz: https://www.youtube.com/watch?v=4Ntd414raYc&list=PL4cUxeGkcC9h60Y8_80q6JerWqsKdAPxn
- [7] *Stack Overflow* [online] Odkaz: <https://stackoverflow.com>
- [8] *GitHub Issues* [online] Odkaz: <https://github.com>
- [9] *Vývoj frontendu* [online] Odkaz: <https://www.youtube.com/watch?v=ukiGFmZ32YA>
- [10] *ChatGPT* [online] Odkaz: <https://chatgpt.com/>