

DELTA – Střední škola informatiky a ekonomie, s.r.o.
Ke Kamenci 151, PARDUBICE

DELTA

Střední škola informatiky a ekonomie
P A R D U B I C E

Maturitní projekt CliSync – Nástroj pro psychology

Příjmení, jméno: KLUCKÝ Jakub
Třída: 4. B
Studijní obor: Informační technologie 18-20-M/01
Školní rok: 2024/2025

1. Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: *Jakub Klucký*
Pro školní rok: *2024/2025*
Třída: *4. B*
Obor: *Informační technologie 18-20-M/01*
Téma práce: *CliSync – Nástroj pro psychology*
Vedoucí práce: *Ing. Tomáš Pešek*

1.1. Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Cílem tohoto projektu je vytvořit webovou aplikaci využívající Next.js pro práci s klienty správu informací, dat, zapisování proběhlé schůzky jednodušší hledání klientů přes filtry a plánování dalších schůzek. Aplikace poskytne psychologům mnohokrát používající papír a tužku zjednodušenou práci s daty klientů, kontaktních informací a přesune vše spojeno s prací na jedno místo a do co nejjednodušší podoby.

1.2. Stručný časový harmonogram (s daty a konkretizovanými úkoly):

1.2.1. 09.2024 - Nastavení projektu a základní struktura:

- Nastavení projektu v Next.js.
- Vytvoření základní struktury front-endu.
- Návrh základního UI/UX rozložení.

1.2.2. 10.2024 - Implementace správy klientů:

- Vývoj rozhraní pro správu informací o klientech.
- Implementace filtrů pro snadné vyhledávání klientů.
- Vytvoření formulářů pro přidávání a úpravu klientských dat.

1.2.3. 11.2024 - Implementace schůzek a rozhraní:

- Vývoj front-endu pro správu schůzek a záznamy o proběhlých schůzkách.
- Přidání možnosti plánování budoucích schůzek (včetně kalendáře).

1.2.4. 12.2024 - Testování a úpravy:

- Testování uživatelského rozhraní (UI) a uživatelské zkušenosti (UX).
- Leštění designu a vylepšení interakce s uživateli.

Prohlašuji, že jsem maturitní projekt vypracoval samostatně, výhradně s použitím uvedené literatury.
V Pardubicích 31.3.2025

2. Poděkování

Děkuji Ing. Tomáši Peškovi za vedení při tvorbě mé maturitní práce a za jeho trpělivost, kterou s námi měl. Rád bych také poděkoval Denisovi Matsenkovi za společné vymyšlení nápadu na tento projekt a za to, že mě naučil vše, co dnes o programování vím. Dále bych chtěl poděkovat psycholožce Mgr Evě Klucké za pomoc při vývoji a testování aplikace. Děkuji a doufám, že Vám bude aplikace přínosem. Nakonec patří mé poděkování rodině, přátelům a mé přítelkyni za jejich morální podporu během celého projektu. Děkuji.

3. Anotace

Práce se zabývá tvorbou frontendové části programu pro psychology. Dokumentace popisuje použité technologie a postupy při tvorbě aplikace. Jako výstup této práce je webová aplikace s názvem „CliSync“

4. Obsah

5. ÚVOD	7
5.1. POJMENOVÁNÍ UŽIVATELŮ APLIKACE	11
6. VYUŽITÉ TECHNOLOGIE	8
6.1. VISUAL STUDIO CODE	8
6.2. MIRO	8
6.3. REACT	8
6.4. NEXTJS	9
6.4.1. <i>Serverside rendering</i>	9
6.5. VERCEL	10
2.1 TAILWIDNCSS.....	10
6.6. DAISY UI.....	10
7. DESIGN APLIKACE.....	11
8. FUNKCIONALITA APLIKACE	11
8.1. KLIENT	11
8.2. TEXTOVÝ EDITOR	14
8.2.1. <i>Základní funkcionalita</i>	14
8.2.2. <i>Sanitace html proti XSS útokům</i>	16
1.1. KALENDÁŘ.....	16
2. TESTOVÁNÍ APLIKACE	20
9. ZÁVĚR.....	21
10. SEZNAM LITERATURY	22
10.1. VISUAL STUDIO CODE (VSCODE):	22
10.2. MIRO:	22
10.3. REACT:	22
10.4. LLM	22
10.5. DATA IN EU	22
10.6. MOST POPULAR FRAMEWORKS.....	22
10.7. MOST POPULAR CODE EDITORS	22
10.8. NEXT.JS:	22
10.9. VERCEL:	22
10.10. VERCEL INFRASTRUCTURE	23
10.11. TAILWIND CSS:	23
10.12. DAISYUI:	23
10.13. XSS ÚTOK:	23
11. SEZNAM OBRÁZKŮ	24

5. Úvod

Aplikace CliSync je moderní nástroj navržený pro zjednodušení administrativní práce psychologů. Mezi psychology nejsou stanovené zásady zapisování informací o klientech. Psychologové často uchovávají data lokálně na počítači, což bývá nezálohované a neorganizované. [3] Stávající pracovní procesy však často zahrnují manuální úkoly, jako je vyhledávání informací o klientech, plánování schůzek v oddělených systémech nebo ruční zapisování poznámek. Tyto činnosti mohou být časově náročné a zbytečně komplikované. Funkčnost aplikace byla posílena integrací umělé inteligence, která napomáhá při zpracování poznámek z konzultací. Umělá inteligence by neměla být používána v aplikacích, kde není její využití opodstatněné. Nicméně v případě práce s textem, například při poznámkách, sumarizaci nebo návrzích na vylepšení textu, představují LLM (Large Language Models) ideální nástroj. Jejich použití v této práci je smysluplné, protože jsou vytrénované na obrovském množství textů dostupných na internetu a dosahují vysoké úrovně přesnosti [5].

Při tvorbě aplikace chceme klást důraz na zabezpečení. Vyvinut bude vlastní způsob autentifikace pomocí JWT tokenů a E2EE (End-to-End Encryption), aby k datům neměl přístup nikdo jiný než psycholog – ani my jako vývojáři aplikace. Klíče pro šifrování budou generovány lokálně a nebudou sdíleny mimo zařízení uživatele. Pro dodržení evropských legislativních standardů je veškerá datová infrastruktura umístěna na území EU. [6] Zabezpečení bude také hlavní důvod integrace self-hosted AI řešení, abychom minimalizovali externí závislosti a zajistili maximální ochranu citlivých dat.

Frontend aplikace CliSync bude zodpovědný za tvorbu uživatelského rozhraní, které bude intuitivní a přehledné. Hlavním cílem je nabídnout uživatelům snadno přístupné funkce, které minimalizují čas strávený administrativními úkoly a umožní jim věnovat více pozornosti samotné práci s klienty. Frontend bude navržen tak, aby bylo možné aplikaci používat bez potřeby školení nebo jiných než základních znalostí práce s počítačem.

Hlavní frontendové části zahrnují:

- **Karta klienta:** Poskytuje rychlý přístup ke klíčovým informacím o klientovi, jako jsou kontaktní údaje, historie schůzek a poznámky.
- **Kalendář schůzek:** Integrace s Google Kalendářem umožňuje snadné plánování a přehled o nadcházejících konzultacích.
- **Optimalizace uživatelského rozhraní:** Minimalistický a responzivní design je přizpůsoben potřebám psychologů, aby mohli aplikaci efektivně využívat na stolních i mobilních zařízeních.

Jako technologie v projektu volím React a Next.js, které umožňují server-side rendering a statickou generaci, což zlepší výkon aplikace, zejména na pomalejších zařízeních. Tyto nástroje také umožní vytvářet znovu použitelné komponenty, což usnadňuje vývoj a údržbu. Navíc mají velkou uživatelskou základnu, [7] což přináší výhody při hledání řešení problémů nebo vylepšování funkcionalit.

Frontend aplikace CliSync se tedy bude zaměřen na to, aby uživatelům poskytl pohodlný a efektivní způsob, jak organizovat svůj pracovní den, komunikovat s klienty a zaznamenávat klíčové informace – vše v jednom uceleném rozhraní.

6. Využité technologie

Při vývoji aplikace CliSync bylo využito několik moderních nástrojů a technologií, které umožnily zajistit výkon, škálovatelnost a bezpečnost aplikace. Následující kapitoly představují klíčové technologie a jejich přínos pro projekt.

6.1. Visual Studio Code

Visual Studio Code [1] je open-source editor pro úpravu kódu vyvinutý společností Microsoft, dostupný pro operační systémy Windows, macOS a Linux. Jedná se o výkonný a flexibilní nástroj, který usnadňuje tvorbu, úpravu a správu kódu. Podporuje mnoho programovacích jazyků, čímž se stával univerzálním řešením. Jednou z jeho klíčových vlastností je IntelliSense, což je funkce automatického doplňování kódu, která nabízí chytré návrhy na základě analýzy projektu. Kromě toho obsahuje integrovaný terminál, který umožňuje spouštět příkazy přímo z editoru, a vestavěné nástroje pro ladění, které pomáhají identifikovat a řešit chyby přímo v kódu. Díky podpoře rozšíření lze přidávat nové funkce, například lintování, a podporu pro specifické technologie a frameworky. VS Code je rovněž dobře integrován s Gitem a GitHubem, což usnadňuje spolupráci v týmu, která byla pro náš projekt potřeba. Oblíbený je díky možnosti vytvořit vlastní rozšíření, kvůli kterým z VS Code vytvořila komunita momentálně nejlepší editor. [8]

6.2. Miro

Miro [7] je online platforma pro vizuální spolupráci, která umožňuje týmům pracovat společně v reálném čase nebo asynchronně. Je známá především jako digitální whiteboard, který nabízí širokou škálu nástrojů pro brainstorming, plánování, designování a řízení projektů. Uživatelé mohou vytvářet diagramy, mapy myšlenek, vývojové diagramy, wireframy a mnoho více. Díky intuitivnímu rozhraní a snadnému sdílení umožňuje spolupráci, a to i na dálku. Aplikace je využívána například vývojáři nebo designery. Její klíčovou vlastností je neomezené plátno, na kterém lze vizualizovat složité nápady, strukturovat informace a propojit různé formáty obsahu, což podporuje kreativitu a týmovou produktivitu.

6.3. React

React [] je open-source JavaScriptová knihovna vyvinutá společností Meta pro tvorbu uživatelských rozhraní. Umožňuje jednoduše a efektivně vytvářet webové aplikace. Především jsem potřeboval nástroj, který umožní vytvářet uživatelské rozhraní efektivně, flexibilně a škálovatelně. React nabízí komponentový přístup, což umožňuje rozdělit aplikaci na menší, znovupoužitelné části, díky tomu je vývoj nejen rychlejší, ale i jednodušejí a udržitelný přehledně.

Dalším důvodem je Virtuální DOM, který zajišťuje výkon aplikace. React provádí aktualizace uživatelského rozhraní pouze tam, kde došlo ke změnám, což je pro moji aplikaci, která zpracovává obsah dynamicky, jako například můj kalendář nebo seznam klientů. Deklarativní přístup Reactu pak usnadňuje definici toho, jak má uživatelské rozhraní vypadat v různých stavech, což je ideální pro práci s komplexními daty a různými scénáři interakce uživatelů. Jeden z hlavních důvodů proč jsem využil react jsou react hooky. Díky hookům, jako je useState nebo useEffect, jsem mohl pracovat se stavem a vedlejšími efekty aplikace jednoduše a bez nutnosti psát třídy, což zrychlilo a zpříjemnilo vývoj.

6.4. NextJS

Next.js [9] je framework postavený nad Reactem, který rozšiřuje možnosti vývoje mé webové aplikace. Nabízí Server-Side Rendering (SSR) a Static Site Generation (SSG), což vede ke zrychlení načítání stránek a lepší SEO optimalizaci. Například dynamické stránky, jako je seznam klientů nebo kalendář, mohou být vykresleny rychleji a vyhledávače je snadněji indexují. Automatický routing na základě struktury souborů v projektu zjednodušuje správu aplikací, zatímco API Routes umožňuje vytvářet backendové funkce přímo v aplikaci bez potřeby samostatného serveru.

Next.js také nabízí **Image Optimization**, která automaticky optimalizuje velikost a formát obrázků, čímž zlepšuje výkon aplikace a uživatelskou zkušenost, zejména na mobilních zařízeních. Kromě toho obsahuje **Middleware**, které umožňuje provádět operace nad requesty a responzy ještě předtím, než jsou zpracovány ve stránkách aplikace. To je užitečné například pro autentifikaci nebo přesměrování uživatelů na základě jejich stavu.

Další významnou funkcí Next.js je **Incremental Static Regeneration (ISR)**, což umožňuje aktualizovat staticky generované stránky bez nutnosti rebuildu celé aplikace. Díky tomu mohou být statické stránky průběžně aktualizovány novými daty, aniž by bylo nutné čekat na kompletní nasazení nové verze webu.

Next.js podporuje také **Edge Functions**, které umožňují běh serverových funkcí přímo na okraji sítě (Edge Network), což vede k extrémně nízké latenci a rychlému zpracování požadavků. To je ideální například pro personalizaci obsahu nebo A/B testování.

Framework také nabízí **Built-in Internationalization (i18n)**, což usnadňuje správu vícejazyčných aplikací bez nutnosti externích knihoven. Vývojářům poskytuje i **Automatic Code Splitting**, čímž minimalizuje velikost načítaných JavaScriptových souborů a zlepšuje výkon.

Protože je postavený na Reactu, poskytuje všechny jeho výhody, včetně hooků, komponentového přístupu a Virtuálního DOMu, ale přidává pokročilé funkce bez nutnosti složité konfigurace.

Navíc má **built-in podporu pro TypeScript**, což zjednodušuje vývoj a zlepšuje stabilitu kódu.

Podporuje také moderní nástroje jako Webpack 5, SWC pro rychlou kompilaci a Rust-based minifikaci kódu.

Použití Next.js je ideální volbou, pokud hledáte robustní řešení pro vývoj rychlých, škálovatelných a optimalizovaných aplikací s vysokým výkonem a výbornou podporou pro SEO.

6.4.1. Serverside rendering

Server-Side Rendering přináší několik zásadních výhod, zejména pro webové aplikace, které vyžadují vysoký výkon, rychlé načítání obsahu a optimalizaci pro vyhledávače. SSR předrenderuje obsah na serveru, takže uživatel dostane kompletní HTML stránku ještě před načtením JavaScriptu. To znamená, že uživatelé uvidí obsah téměř okamžitě, což zlepšuje uživatelskou zkušenost. Je to zvláště výhodné pro aplikace s obsahem, který se často mění, jako jsou e-shopy, zpravodajské portály nebo dynamické tabulky dat. Dalším důvodem pro použití SSR je lepší SEO. Vyhledávače upřednostňují weby, které servírují plný obsah ve formě HTML, což usnadňuje indexaci a zlepšuje viditelnost webu. SSR také umožňuje generovat personalizované stránky přímo na serveru, například zobrazení uživatelsky specifických dat nebo přizpůsobení obsahu na základě lokalizace či zařízení. Další výhodou je lepší výkon na pomalejších zařízeních, protože většina zátěže je přesunuta na server. SSR lze navíc kombinovat s dalšími strategiemi, jako je Static Site Generation pro statické stránky nebo Client-Side Rendering pro interaktivní prvky. Tato flexibilita umožňuje optimalizovat aplikaci pro různé

scénáře, například pro real-time aktualizace obsahu, kdy server načte aktuální data při každém požadavku, a uživatel tak vždy dostane nejnovější verzi. SSR je vhodné, pokud potřebujete lepší SEO, pracujete s dynamickým obsahem, chcete snížit dobu načítání stránky nebo podporovat méně výkonná zařízení.

6.5. Vercel

Vercel [10] je cloudová platforma pro naazování a škálování moderních webových aplikací. Zaměřuje se na rychlost, jednoduchost a podporu vývojářů

Používání platformy Vercel přináší řadu výhod, které usnadňují vývoj a nasazení webových aplikací, zejména těch postavených na frameworku Next.js. Vercel je navržen tak, aby vývojáři mohli rychle nasazovat své projekty bez složité konfigurace serverů nebo infrastruktury. Jednou z hlavních výhod je možnost nastavení a automatického nasazení z GitHub repozitářů. Při každé změně kódu je aplikace automaticky nasazena na produkční i testovací prostředí. Dalším důvodem je vysoký výkon a optimalizace pro koncové uživatele. Vercel poskytuje globální síť CDN, která zajišťuje rychlé načítání stránek po celém světě.[11] Platforma také nativně podporuje server-side rendering a static site generation, což je perfektní kombinace pro aplikace založené na Next.js. Navíc nabízí funkce jako edge middleware, které umožňují přizpůsobení obsahu přímo na okraji sítě, čímž se zlepšuje uživatelská zkušenost. Velkou výhodou je také jednoduchost a přívětivé rozhraní, které umožňuje monitorovat stav nasazení, výkon aplikace nebo správu domén. Vercel navíc podporuje týmovou spolupráci, což znamená, že více vývojářů může pracovat na jednom projektu efektivněji. Pro mě je Vercel ideálním nástrojem díky své jednoduchosti, výkonu a bezproblémové integraci s moderními webovými technologiemi.

2.1 TailwindCSS

TailwindCSS [12] je framework na stylování aplikací který používá utility přístup. Místo psaní vlastních tříd umožňuje používat předdefinované utility classes přímo v html. Tyto třídy reprezentují konkrétní stylové vlastnosti, jako jsou barvy, mezery, velikosti fontů, zarovnání nebo stíny. Tailwind také obsahuje nástroj pro optimalizaci, který zajišťuje, že ve výsledném balíčku jsou zahrnuty pouze třídy, které reálně používáte, což zlepšuje výkon aplikace.

6.6. Daisy UI

Daisy UI [13] je plugin který poskytuje vlastní komponenty vytvořené pomocí frameworku Tailwind CSS tím zajišťují možnost znovu použít jednotlivé komponenty a zároveň zachová možnost plně upravit přidáním vlastních Tailwind CSS tagů. Toto bylo klíčové, aby k tvorbě aplikace nebyl potřeba design, ale mohl jsem ji vytvořit sám bez bližších znalostí grafického designu.

7. Design aplikace

Design aplikace představuje proces návrhu a realizace vzhledu a fungování softwarové aplikace. Byl vytvořen tak, aby uživatelé mohli aplikaci používat bez potřeby zúčovacího kurzu. Vše je tvořeno velmi výraznými kontrastními barvami, které zajišťují snadnou orientaci a dobrou čitelnost.

Pro jednotnost designových prvků jsem využil Daisy UI. Tento nástroj mi umožnil rychle a efektivně vytvořit designové komponenty, které jsou snadno přizpůsobitelné a vizuálně konzistentní s požadavky aplikace.

8. Funkcionalita aplikace

8.1. Pojmenování uživatelů aplikace

V dokumentaci jsou pojmenováni uživatelé aplikace a lidé s nimi spojení. Jako uživatel je považován psycholog, jako klient nebo pacient je považován člověk docházející k danému psychologovi na konzultace

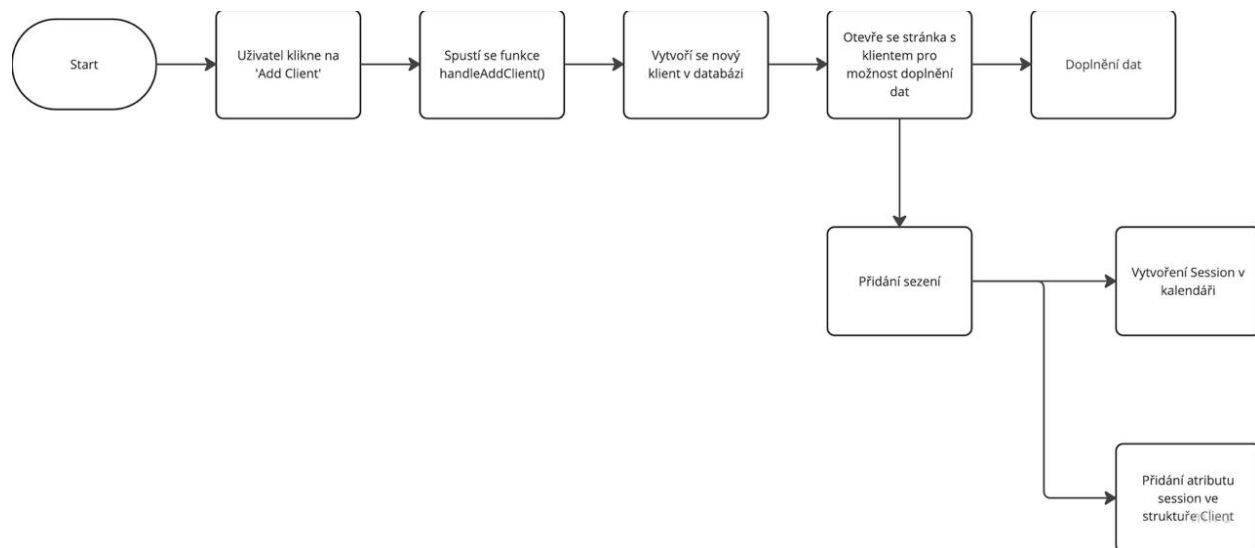
8.2. Klient

Klient obsahuje následující atributy:

- **id:** Unikátní identifikátor klienta,
- **name:** Jméno klienta,
- **email, phoneNumber, age:** Kontaktní informace,
- **quickNotes:** Pole pro rychlé poznámky, které může obsahovat textové shrnutí schůzek nebo důležité body konzultací,
- **attachments:** Soubory připojené k záznamu klienta (například dokumenty nebo obrázky),
- **sessions:** Záznamy schůzek s klientem.

Každá session je propojena s událostí v kalendáři, což zajišťuje přehlednost a snadnou navigaci mezi minulými i budoucími schůzkami. Při změně data nebo času události se příslušný záznam v atributu sessions automaticky aktualizuje. Tento přístup byl vyvinut na základě konzultací s psychology, kteří zdůraznili, že pro přípravu na další schůzky je obvykle dostačující zápis z předchozího sezení.

Diagram níže znázorňuje průběh akce vytvoření klienta



Obrázek 1 Diagram vytvoření Klienta

🏠

📅

📖

Save

Delete

Jan Novák

client id: eae9e104-6263-4732-8007-4966f0c5ba93

Note about client..

Name *

Jan Novák

Email

email

Phone number

phone number

Age

Quick notes

Add note

Attachments

+

Sessions

Create session in calendar

2025.01.17, 20:00 - 21:00

...

💡

?

⚙️

📄

Začátek nové kapitoly

Dnes se klient svěřil, že již dlouho přemýšlí, že by začal dělat sport zvaný hobbyhorsing, rozhodl se tento krok udělat.

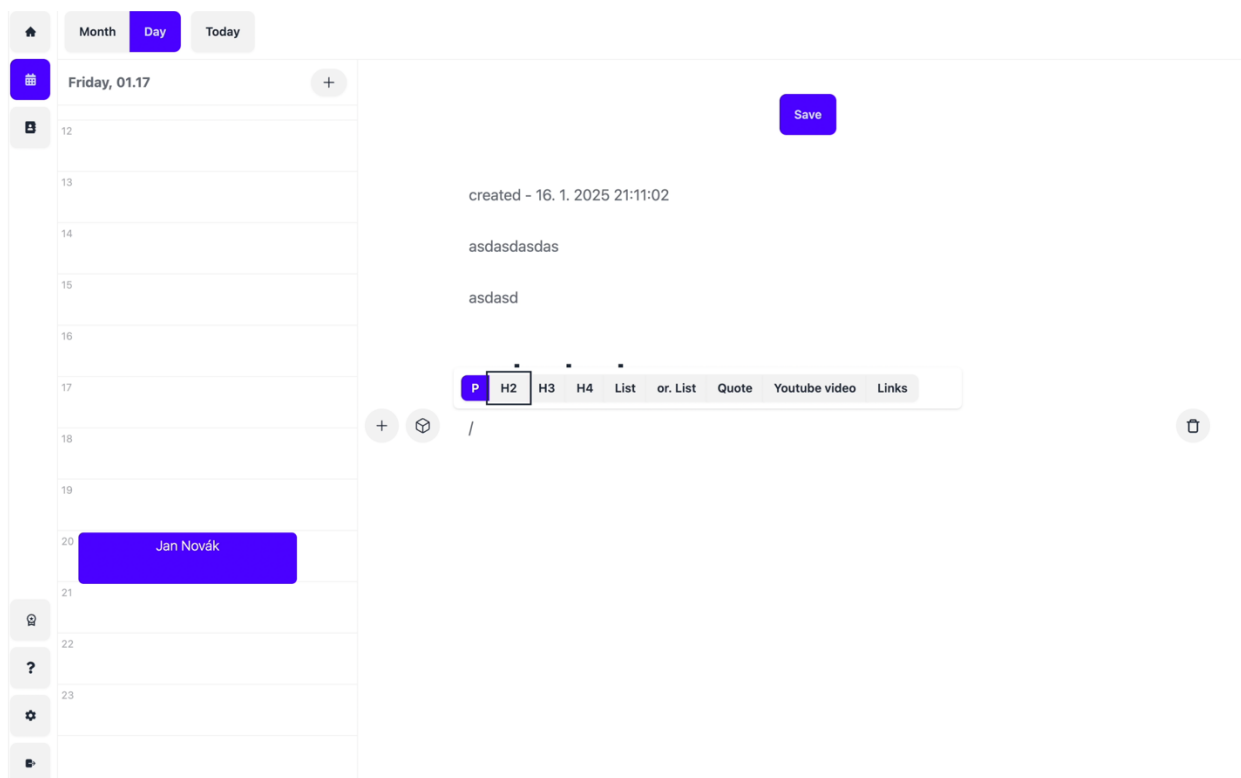
Obrázek 2 Stránka s informacemi o klientovi

8.3. Textový editor

Pro aplikaci jsem vytvořil vlastní textový editor, který zjednodušuje zápis poznámek ze sezení s klienty. Editor je **block scoped** (blokový), což znamená, že každá poznámka je rozdělena do samostatných bloků. Každý blok má svůj specifický typ, například paragraf, nadpis, podnadpis, obrázek atd.

8.3.1. Základní funkcionalita

- **Automatický základní blok:** Při vytvoření nové **session** se automaticky v editoru vytvoří první blok typu paragraph, který je určený pro psaní prostého textu. Současně se spustí funkce `setShowTools(true)`, která nastaví proměnnou `showTools` na `true`. Pokud je `showTools` nastaveno na `true`, zobrazí se u aktuálně aktivního bloku nástroje, které umožňují změnu typu bloku.
- **Vkládání nových bloků:** Uživatel může přidat nový blok buď stisknutím tlačítka Enter, nebo kliknutím na tlačítko +. Nový blok je opět typu paragraph. Speciálně u bloků typu list a ordered-list stisknutí Enter nezpůsobí vytvoření nového bloku, ale spustí funkci `handleEnterKey()`. Ta dynamicky přidá další komponentu `<OrderedListComponent />` do aktuálního `<OrderedList />`.
- **Automatické přizpůsobení výšky bloku:** Každý blok typu `HTMLTextAreaElement` má na začátku pevně stanovenou výšku. Pokud uživatel píše a kurzor dosáhne konce bloku, aktivuje se funkce `adjustHeight` (přes `useEffect()`), která přizpůsobí výšku bloku aktuální velikosti obsahu (`current.scrollHeight`).



Obrázek 3 Textový editor, typy bloků

8.3.2. Sanitace html proti XSS útokům

Vzhledem k tomu, že aplikace je napsaná v Reactu a využívá DOM-based přístup, je zranitelná vůči XSS útokům (Cross-Site Scripting). XSS útok spočívá ve vložení škodlivého JavaScriptu do dynamické webové stránky, což může vést například ke změně adresy stránky nebo přesměrování uživatele.

Aby tomuto bylo předejito:

1. **Sanitace vstupů:** Při generování bloků používám knihovny `sanitize` a `htmlparser2` z Node.js. Tyto nástroje zajistí, že v HTML kódu vytvořeném v blocích nebudou žádné nežádoucí nebo škodlivé prvky.
2. **Bezpečnostní ošetření dat:** Data z editoru jsou ukládána a zobrazována bezpečně, aby se minimalizovalo riziko, že by uživatelský vstup mohl způsobit neočekávané chování aplikace.

Dále bylo nutné v textovém editoru implementovat možnost znovu přepsat blok, který již byl vytvořen proto byla pro každý blok implementována funkce `setFocusedId`, která když uživatel znovu klikne na blok, který už byl vytvořen nastaví proměnou `focusedId` na id bloku, na který bylo kliknuto a umožní znovu upravovat

V práci byl vytvořen vlastní textový editor z těchto důvodů:

- Plynulost práce podobná s editorem v aplikaci Notion.
- Rychlost a jednoduchost zápisu.
- Neexistující opensource řešení vyhovující požadavkům.

1.1. Kalendář

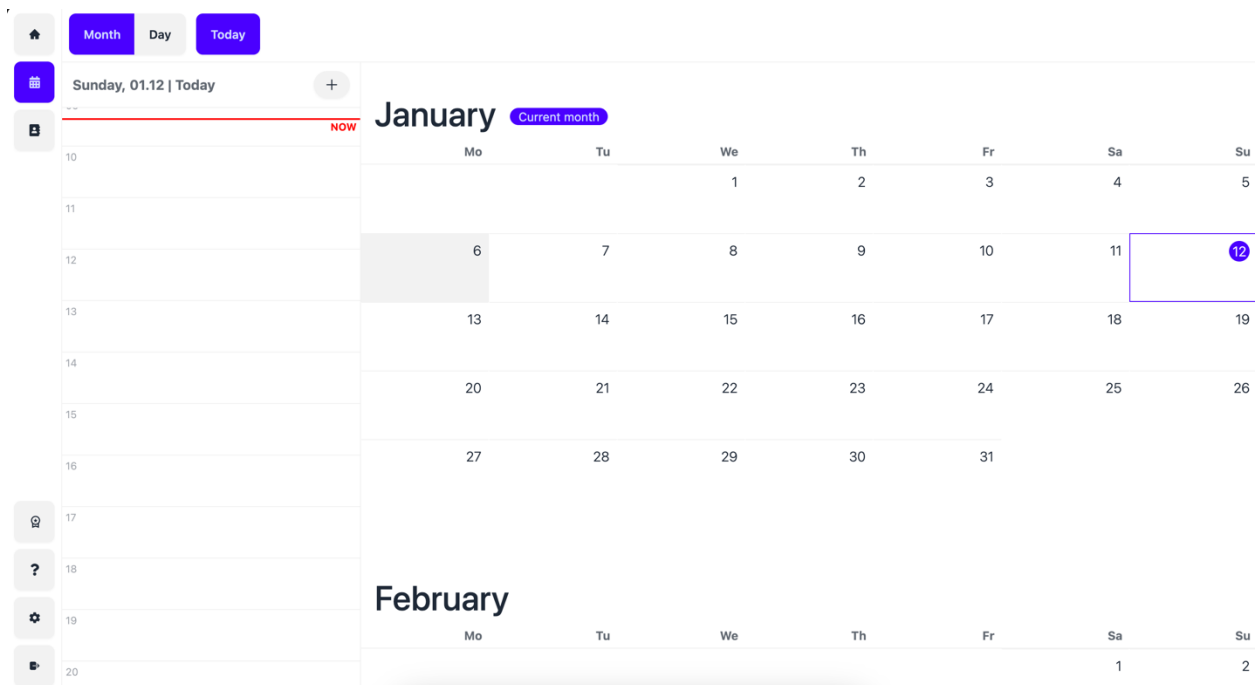
Pro plánování sezení s klienty jsem vytvořil kalendář, který umožňuje snadné přidávání sezení a jejich propojení s Google Kalendářem. To poskytuje uživatelům pohodlný způsob synchronizace pracovního harmonogramu s osobním životem. Kalendář podporuje tři možnosti zobrazení – denní a měsíční pohled, přičemž jednotlivá sezení se automaticky propojují se zápisy, což usnadňuje jejich zpětné dohledání podle data.

Struktura kalendáře je založená na poli naplněném objekty **Day**, které uchovávají veškerá data spojená se sezeními a identifikátor klienta. Každý objekt **Day** se zároveň ukládá do databázové tabulky **sessions**. Pokud má uživatel propojený svůj Google účet, objekt **Day** obsahuje také **GoogleCalendarEvent**, což umožňuje pomocí **Google Calendar API** získávat a zobrazovat data z osobního kalendáře přímo v pracovním kalendáři aplikace. Tímto způsobem se vždy načítají události pro aktuální měsíc, předchozí měsíc a měsíc následující, což zajišťuje plynulou práci s aplikací.

Při návrhu uživatelského rozhraní byla snaha přizpůsobit rozvržení kalendáře tomu, co mně osobně při práci nejvíce vyhovuje. Proto jsem implementoval detailní denní zobrazení na pravé straně a měsíční přehled na levé. Navigace mezi měsíci funguje na principu **nekonečného posouvání (infinite scroll)**. Tento mechanismus využívá předem počítanou výšku vykresleného měsíce, a pokud uživatel posouváním dosáhne jeho konce, aplikace automaticky zavolá funkci **loadNextMonth()** nebo **loadPrevMonth()**, která načte další data do kalendáře. Tento způsob zajišťuje plynulé procházení kalendáře, přičemž v paměti je vždy uchován jeden měsíc navíc oproti aktuálně zobrazenému.

Při implementaci jsem se zabýval i otázkou efektivitu tohoto přístupu ve srovnání s tradiční navigací pomocí tlačítek se šipkami. Během práce bylo zjištěno, že obě metody by vyžadovaly

stejnou logiku načítání dat, takže infinite-scroll je vhodnější, protože nabízí uživatelům plynulejší a přirozenější navigaci bez nutnosti ručního přepínání mezi měsíci.



Obrázek 4 Kalendář - zobrazení měsíce

The image shows a 'Create new session' modal form. The background is a calendar for January 2025, with days of the week (Mo, Tu, We, Th, Fr) and dates (3, 10, 17, 24, 31) visible. The modal form has the following sections:

- Create new session** (Title)
- Time ***: Two date-time pickers showing '1/16/2025, 12:00:00 AM'.
- Client ***: A search bar with a magnifying glass icon and the text 'Search for a client...'.
- Note**: A text area with a placeholder 'e.g. "Client is coming with a friend"'.
- Other**: Two toggle switches.
 - 'Add to my Google Calendar' (currently off).
 - 'Invite client on Google Calendar event' (currently off).
- Create session** (Button)

A warning message is displayed below the 'Note' section: You need to connect your Google Calendar to use this feature. [Visit settings.](#)

Obrázek 5 Kalendář - vytvoření schůzky

2. Testování aplikace

V průběhu tvorby aplikace probíhal spolupráce s psycholožkou, která průběžně testovala aplikaci a poskytovala zpětnou vazbu na funkcionalitu co je potřeba přidat a velké množství bugů, které se v aplikaci nacházely. Tato spolupráce byla přínosná také, protože pomáhala dívat se na implementaci funkcí aplikace z pozice lidí kteří nejsou zdatní v používání počítačů.

9. Závěr

Byla vytvořena funkční aplikace, která má za cíl usnadnit práci psychologům, kteří jsou v současnosti velmi vyhledávanou profesní skupinou. V České republice je psychologů nedostatek a rostoucí poptávka vede k dlouhým čekacím lhůtám na terapii, které mohou dosahovat až tří měsíců. Proto bylo v rámci vývoje aplikace absolvováno několik konzultací s psychology, za účelem zjistit, co by jim pomohlo pracovat efektivněji a tím umožnilo za stejný čas pomoci více lidem.

Aplikace obsahuje všechny plánované funkcionality, které byly na základě zpětné vazby identifikovány jako klíčové pro zjednodušení práce psychologů. Hlavním cílem bylo minimalizovat čas strávený opakovaným vyhledáváním telefonních čísel, starých zápisů nebo jiných informací z předchozích sezení.

Při implementaci aplikace byl vytvořen kalendář, který je propojen s plánovanými schůzkami navázanými na samotné klienty, karta klienta, pro přehledné zobrazení dat. Implementován byl textový editor optimalizovaný pro rychlé a efektivní psaní zápisů ze sezení. Vzhledem k citlivosti zpracovávaných údajů bylo implementováno dvoufaktorové ověření a end-to-end šifrování, pro zajištění maximální bezpečnosti klientských dat.

V budoucnu plánujeme aplikaci rozšířit o nové funkce, jako je podpora mobilních zařízení formou nativní aplikace pro iOS a Android. Zvažujeme také integraci dalších nástrojů, například přímé komunikace s klienty nebo pokročilých analytických funkcí pro sledování vývoje jednotlivých případů. Další rozšíření může zahrnovat rozšíření AI funkcí, jako je predikce potřeb klientů na základě záznamů nebo generování podrobných plánů terapií na míru.

Jako technologie jsem zvolil React a Next.js, které umožňují server-side rendering a statickou generaci, což zlepšuje výkon aplikace, zejména na pomalejších zařízeních. Tyto nástroje také umožňují vytvářet znovu použitelné komponenty, což usnadňuje vývoj a údržbu. Navíc mají velkou uživatelskou základnu, [7] což přináší výhody při hledání řešení problémů nebo vylepšování funkcionalit.

Frontend aplikace CliSync se tedy zaměřuje na to, aby uživatelům poskytl pohodlný a efektivní způsob, jak organizovat svůj pracovní den, komunikovat s klienty a zaznamenávat klíčové informace – vše v jednom uceleném rozhraní.

10. Seznam literatury

10.1. Visual Studio Code (VSCode):

Microsoft. *Visual Studio Code*. [online]. [cit. 2025-01-19]. Dostupné z: <https://code.visualstudio.com/>

10.2. Miro:

Miro. *Miro: Online Whiteboard for Visual Collaboration*. [online]. [cit. 2025-01-19]. Dostupné z: <https://miro.com/>

10.3. React:

Meta Platforms, Inc. *React – A JavaScript library for building user interfaces*. [online]. [cit. 2025-01-19]. Dostupné z: <https://react.dev/>

10.4. LLM

Originality.ai What LLM is the most accurate. Online. Originality.ai. 2024. Dostupné z: <https://originality.ai/blog/what-llm-is-the-most-accurate> [cit. 2025-02-18].

10.5. Data in EU

Europa EU. Online. Europa.eu. 2024. Dostupné z: https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/index_en.htm. [cit. 2025-02-27].

10.6. Most popular frameworks

Statista. Online. Statista.com. 2024. Dostupné z: <https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>. [cit. 2025-02-27].

10.7. Most popular Code editors

Stack Overflow. Online. Stackoverflow.co. 2024. Dostupné z: <https://survey.stackoverflow.co/2024/technology#most-popular-technologies-new-collab-tools>. [cit. 2025-02-27].

10.8. Next.js:

Vercel. *Next.js: The React Framework*. [online]. [cit. 2025-01-19]. Dostupné z: <https://nextjs.org/>

10.9. Vercel:

Vercel. *Vercel: Develop. Preview. Ship*. [online]. [cit. 2025-01-19]. Dostupné z: <https://vercel.com/>

10.10. Vercel Infrastructure

Vercel Infrastructure. Online. Stackoverflow.co. 2023. Dostupné z: <https://vercel.com/blog/behind-the-scenes-of-vercel-infrastructure>. [cit. 2025-02-27].

10.11. Tailwind CSS:

Tailwind Labs. *Tailwind CSS: Rapidly build modern websites without ever leaving your HTML*. [online]. [cit. 2025-01-19]. Dostupné z: <https://tailwindcss.com/>

10.12. DaisyUI:

DaisyUI. *DaisyUI: Tailwind CSS Components*. [online]. [cit. 2025-01-19]. Dostupné z: <https://daisyui.com/>

10.13. XSS útok:

OWASP Foundation. *Cross Site Scripting (XSS)*. [online]. [cit. 2025-01-19]. Dostupné z: <https://owasp.org/www-community/attacks/xss/>

11. Seznam Obrázků

Obrázek 1 Stránka s informacemi o klientovi.....	13
Obrázek 2 Textový editor, typy bloků	15
Obrázek 3 Kalendář - zobrazení měsíce	18
Obrázek 4 Kalendář - vytvoření schůzky.....	19