

# **Gymnázium, Praha 6, Arabská 14**

*Obor Programování*

## **Maturitní práce**



Jakub Švéda

## **Skautská sociální platforma**

Březen 2024

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, Arabská14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V ..... dne .....

Podpis .....

**Název práce:** Skautská sociální platforma

**Autor:** Jakub Švéda

**Abstrakt:**

Cílem je vytvoření responzivní webové aplikace pro skauty, skrz kterou budou moci komunikovat v rámci jednotlivých částí struktury skautského střediska. Mimo jiné budou moci skrze aplikaci domlouvat akce a události. V případě schválení od českého Junáka bude využívat jednotný skautský účet pro přihlašování (SkautIS), v opačném scénáři pak bude mít aplikace přihlašování vlastní.

The aim is to create a responsive web application for scouts to communicate within the different parts of the scout centre structure. Among other things, they will be able to arrange events through the app. If approved by the Czech Junák, it will use a single scout account for logging in (SkautIS), otherwise the application will have its own authentication logic.

# Obsah

1.	Úvod.....	5
2.	Použité technologie .....	6
2.1	Next.js.....	6
2.2	Typescript .....	6
2.3	Material UI.....	7
2.4	React.js .....	7
2.5	PostgreSQL .....	7
2.6	Socket.io .....	7
2.7	Zod.....	8
2.8	Crypto.js .....	8
2.9	Axios .....	8
3.	Frontend .....	9
3.1	Design komponent .....	9
3.2	Stránky projektu .....	9
3.2.1	Domovská stránka - / .....	9
3.2.2	Zprávy - /messages .....	10
3.2.3	Chat - /messages/chat/{id skupiny}.....	10
3.2.4	Plány - /plans.....	10
3.2.5	Událost - /plans/event/{id události} .....	10
3.2.6	Přihlašovací stránka - /auth/signin .....	10
3.2.7	Registrační stránka - /auth/signup.....	11
4.	Backend .....	12
4.1	Ověření uživatelů a zabezpečení.....	12
4.2	Chatování .....	13
4.3	Ukládání médií.....	14
4.4	Vytváření událostí .....	14
4.5	Databáze.....	15
5.	Návod k instalaci.....	16
6.	Závěr .....	18
7.	Bibliografie.....	19

# 1. Úvod

Cílem maturitní práce je vytvořit webovou aplikaci pomocí technologie Next.js na bázi programovacího jazyku Typescript tak, aby měla responzivní design a byla snadná na používání i pro méně technicky zdatné uživatele.

Aplikace bude sociální sítí pro skauty. Budou skrz ni moci sdílet fotky, videa a komunikovat jak v rámci oddílů, tak i samotného střediska. Dále bude obsahovat možnost zobrazení základních informací o střediscích a o oddílech. Skrze aplikaci bude také možné domlouvat termíny různých akcí (schůzky, rady).

V případě schválení od českého Junáka bude využívat jednotný skautský účet pro přihlašování (tzv. SkautIS), v opačném scénáři pak bude mít aplikace přihlašování vlastní se základními bezpečnostními prvky.

## 2. Použité technologie

V této kapitole jsou popsány všechny zásadní technologie, jež byly v projektu použity. V některých případech jsou zde taktéž vysvětlené důvody, které byly rozhodující při jejich výběru.

### 2.1 Next.js

Next.js [1] je open-source frameworkem pro vývoj webových aplikací od společnosti Vercel. Umožňuje vytvářet backendové části webových aplikací pomocí knihovny *React.js*. Mezi hlavní benefity, které tento framework přináší, patří funkce statického a dynamického renderování, jež zlepšují efektivitu a rychlost aplikací postavených právě na knihovně *React.js*.

Dynamické renderování se využívá ke zrychlenému vykreslování samotné stránky na straně klienta. To se může hodit například pro jednodušší záznam obsahů stránek pro vyhledávací nástroje vyhledávačů, které potřebují získat obsah stránky co možná nejrychleji.

Naopak statické renderování absolutně nahrazuje generování stránky na straně klienta, čímž dojde k bezpečnějšímu fungování aplikace a také k tomu, že není třeba generování stejného obsahu, který se často nemění, vícekrát, než je potřeba. Stránka se totiž předgeneruje pouze jednou a následně už jen při případných změnách.

### 2.2 Typescript

Typescript [2] je programovacím jazykem velmi podobný Javascriptu. Funguje jako jeho nadstavba obohacená o kontrolu typů proměnných. Mimo základní typy, jež jsou obvyklé ve standardních programovacích jazycích používajících typy proměnných, lze též definovat vlastní typy (tzv. interfaci). To přináší takovémuto projektu lepší přehlednost kódu, chytání typových chyb a jiné výhody, jež vedou k přesnějšímu a méně chybovému vývoji webových aplikací.

## 2.3 Material UI

Material UI [3] je open-source knihovna komponent pro React.js, která implementuje design společnosti Google. Knihovna implementuje jednotný design, jež je možné se znalostí kaskádových stylů (CSS) jednoduše modifikovat dle specifických požadavků vlastní webové stránky.

## 2.4 React.js

React.js (dále jen React) [4] je frontendová knihovna od společnosti Meta. Je postavená na vytváření komponent, tedy blocích kódu, které plní určitou funkci a sami se starají o obnovu svých dat. Komponenty zároveň kombinují HTML a Javascript (v případě tohoto projektu Typescript), což vede ke vzniku jakýchsi samostatně fungujících widgetů a zároveň umožňuje HTML dynamicky vykreslovat podle aktuálních podmínek (stavů) aplikace.

## 2.5 PostgreSQL

Pro ukládání dat byl v projektu použit open-source relační databázový systém PostgreSQL [5]. Byl zvolen zejména proto, že patří k nejrobustnějším, nejefektivnějším a zároveň bezplatným databázovým systémům. Podporuje všechny základní důležité prvky databáze, jimiž jsou například ACID transakce.

## 2.6 Socket.io

Socket.io [6] je knihovna umožňující komunikaci mezi webovými klienty (prohlížeči) a servery v reálném čase. To znamená, že klient i server si mohou navzájem posílat data v reálném čase, což následně umožňuje vytvářet webové aplikace, které neustále reagují na změny, aniž by bylo nutné neustále obnovovat stránku. V tomto projektu je konkrétně použita k chatování mezi různými skupinami.

Socket.io funguje na principu webového socketu, což je speciální typ síťového připojení, které umožňuje obousměrnou komunikaci v

reálném čase. Pokud prohlížeč nepodporuje webové sockety, Socket.io použije záložní metody, aby se zajistila kompatibilita s většinou prohlížečů.

## 2.7 Zod

Zod [7] je open-source knihovna pro validaci dat. Umožňuje definovat přesná schémata pro standardní datové struktury a poskytuje nástroje pro jejich validaci, parsování a manipulaci. Je snadná na použití a své využití najde zejména při ověřování dat v REST API v backendových částech aplikací. Tímto způsobem byla také využita v tomto projektu.

## 2.8 Crypto.js

Crypto.js [8] je open-source knihovna JavaScript (zde použita s typy pro Typescript), která poskytuje implementace široké škály standardních a bezpečných kryptografických algoritmů. Umožňuje přidávat funkce jako šifrování, dešifrování, hašování a kódování do formátu Base64.

## 2.9 Axios

Axios [9] je populární open-source knihovna pro JavaScript, která se používá k provádění HTTP požadavků (například GET, POST, PUT a DELETE) jak na straně klienta, tak i na straně serveru. Hlavní výhodou je zjednodušení http požadavků na server, které vězí například v parsování dat ve formátu JSON.



## 3. Frontend

### 3.1 Design komponent

Pro design drtivé většiny komponent byla v projektu použita knihovna Material UI s širokou škálou různých komponent [10]. Díky této knihovně drží aplikace konstantní vzhled, co se týče barev, velikostí odsazení a podobně, což ve výsledku vede k dodržení základních pravidel vzhledu a k tomu, že tento vzhled dává dohromady smysl. Zároveň dělá webovou aplikaci responzivní pro zařízení od nejmenších telefonů, až po velké kancelářské monitory.

Hlavní výhodou Material UI je prvek nazývaný téma. Téma definuje základní parametry jako je primární barva, sekundární barva, odsazení a jiné. To vše je zapsáno v jednom souboru (zde v souboru *theme.tsx*) a ovlivňuje při vykreslení všechny podporované komponenty.

Další využití v projektu našla možnost úpravy designu komponent dle vlastních preferencí pomocí objektu schovaném v atributu *sx* jednotlivých komponent. Na rozdíl od tématu mění takovýto objekt pouze danou komponentu nikoliv i všechny ostatní, které jsou ovlivněny stejným typem vzhledu. Ke změnám využívá standardní atributy tříd jazyka CSS.

### 3.2 Stránky projektu

V nynější kapitole se nachází stručný popis jednotlivých stránek, jejich URL adresa a předpoklady nutné k jejich přístupu.

#### 3.2.1 Domovská stránka - /

Na domovské stránce se má vyskytovat základní přehled novinek pro konkrétního uživatele. Mělo by zde být informace o uživateli, novinky z chatů, knihovny fotek a plánů. V aktuální verzi však tato data nejsou k dispozici. Bez přihlášení nelze ke stránce přistoupit.

### 3.2.2 Zprávy - /messages

Na stránce */messages* se vyskytuje seznam chatů, který lze třídit dle skupin (priorit uživatele). Po rozkliknutí některého z chatů je uživatel přesměrován do daného chatu. Bez přihlášení nelze ke stránce přistoupit.

### 3.2.3 Chat - /messages/chat/{id skupiny}

Na této stránce lze nalézt konkrétní chat dle id skupiny tohoto chatu. To musí být skryté, aby někdo cizí nemohl na stránku přistoupit a posílat do chatu vlastní (nevyžádané) zprávy. Z toho důvodu id tvoří specificky vytvořený hash. Bez přihlášení a práv do konkrétního chatu nelze ke stránce přistoupit.

### 3.2.4 Plány - /plans

Na stránce */plans* se vyskytuje seznam plánů neboli událostí. Lze pomocí stránky též události vytvářet. Po rozkliknutí některého z plánů je uživatel přesměrován na stránku s konkrétnějšími údaji. Bez přihlášení nelze ke stránce přistoupit.

### 3.2.5 Událost - /plans/event/{id události}

Na této stránce lze nalézt konkrétní událost dle id události. Záleží na stavu dané události. Pokud je hlasování o termínu akce ukončeno, pak se zobrazí pouze informace finální, tedy bez možnostmi voleb. Bez přihlášení a práv pro zobrazení dané události nelze ke stránce přistoupit.

### 3.2.6 Přihlašovací stránka - /auth/signin

Na přihlašovací stránce je k nalezení jednoduchý formulář pro přihlášení do aplikace. Po přihlášení je klient přesměrován na domovskou stránku aplikace.

### 3.2.7 Registrační stránka - /auth/signup

Na registrační stránce se nachází formulář pro registraci do aplikace. Formulář je schopen zobrazit případné nedostatky po odeslání požadavku. Po následném úspěšném zaregistrování je klient přesměrován stejně jako po přihlášení na domovskou stránku aplikace.

## 4. Backend

V této kapitole jsou popsány základní vlastnosti serverové části aplikace, jejich částečné fungování a úskalí, která řeší.

### 4.1 Ověření uživatelů a zabezpečení

Přihlášení, registrace a odhlášení patří k základním vlastnostem většiny webových aplikací, které využívají data jednotlivých uživatelů. Díky tomuto způsobu lze tvořit různorodé prostředí pro různé uživatele, kteří mají v drtivé většině též jiná práva. Jinak tomu není ani v této aplikaci.

Na straně serveru dochází zprvu k registraci uživatele s obvyklými atributy, jimiž jsou uživatelské jméno, přezdívka, email a v neposlední řadě také hash hesla. K vytvoření hashe pro každého uživatele využívá aplikace algoritmus SHA512. Přestože v dnešní době je obvyklejší jeho slabší verze SHA256, aplikace využívá tento typ s myšlenkou na budoucí vývoj počítačů schopných starší verzi prolomit.

Pro bezpečnější práci s hesly je použit hash v kombinaci se solí [11], tedy způsobu, jak ještě zvýšit originalitu každého hashe hesla i v případech, kdy mají různí uživatelé stejné heslo. To zajišťuje odolnost aplikace pro některým typům hackerských útoků. Hash je v databázi uložen ve tvaru „sůl%hash“, to pro případ, kdyby došlo v budoucnu ke změnám hashovacího algoritmu. Staří uživatelé se starým algoritmem by stále mohli k aplikaci přistoupit.

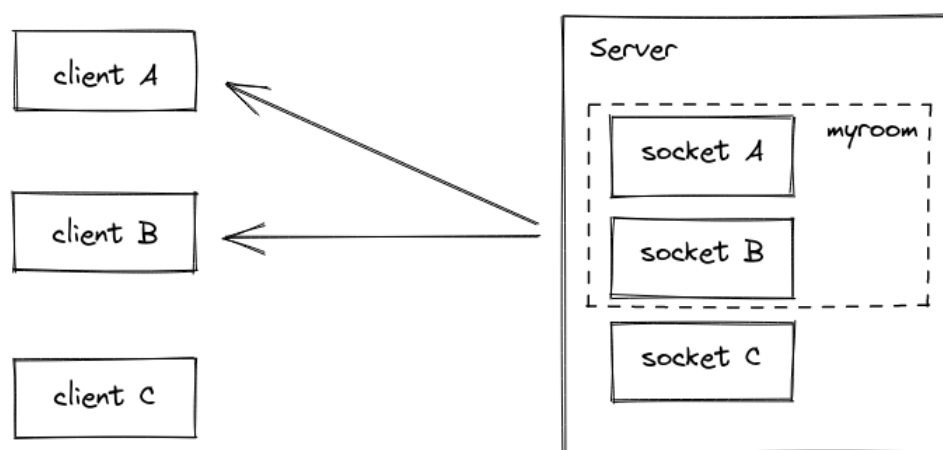
Po přihlášení je uživateli vytvořen univerzální token, který se stává jeho vnitřní identitou pro přístup k datům z REST API aplikace. Token je vrácen uživateli po přihlášení ve formě cookie, která se při specifických akcích sama obnovuje tak, aby mohl uživatel pohodlně pracovat a nebyl náhodně odhlášen.

## 4.2 Chatování

Jednou z nejpodstatnějších částí a zároveň zřejmě nejnáročnějších částí aplikace je možnost chatování, konkrétně tedy posílání zpráv a fotek skrze chat v reálném čase.

Tou nejnáročnější částí je právě real-time odezva. Existují celkem dva základní způsoby, jakým ji zajistit. Prvním je, že klient v opakovaných intervalech posílá na server plnohodnotné http/s dotaz, zda nebyla jiným uživatelem zveřejněna nějaká zpráva. Tento způsob je ale velice náročný a jediným způsobem, jak ho zefektivnit je zvětšení intervalu pro posílání dotazů. To bohužel vede k tomu, že zprávy již nepřichází v reálném čase, tedy jeden z předpokladů funkčnosti nebude splněn.

Lepším řešením je využití socketů pomocí knihovny Socket.io. Uživatelé spolu komunikují díky pomocnému serveru v souboru *server.js* [12], který zajišťuje právě jen a pouze real-time komunikaci. Server nijak nepracuje s relační databází, ale jen posílá zprávy uživatelům, kteří jsou ve stejných skupinách. Ukládání zpráv je totiž zajištěno pomocí plnohodnotných dotazů na REST API serveru.



*Diagram fungování knihovny Socket.io - 1*

Na obrázku [13] výše lze vidět, jak knihovna *Socket.io* dělí klienty do komunikačních skupin. Jak tedy samotná komunikace funguje?

Prvotně po přistoupení ke stránce s chaty se ověří, zda daný uživatel má do této skupiny přístup. Následně se načtou poslední zprávy z chatu. V tuto chvíli se zároveň klient připojí do skupiny, kterou *Socket.io* vytvoří. Od této chvíle klient jen poslouchá, zda přichází nějaké zprávy od jiných uživatelů. Pokud pošle zprávu, tak vykoná 2 akce. Prvně pošle tuto zprávu do skupiny knihovny *Socket.io* na pomocném serveru. Tím se zajistí komunikace v reálném čase. Nyní je však ještě potřeba tuto zprávu uložit do databáze, aby i jiní uživatelé, kteří přistoupí do chatu později, mohli zprávy načíst. Pošle se tedy dotaz na server, který tuto zprávu uloží.

### 4.3 Ukládání médií

Další výzvou projektu bylo zvolit vhodný způsob ukládání médií. Neboť se jednalo o ukládání velkých souborů jako jsou například fotky, bylo nejlepší řešením vytvoření jakési zjednodušené verze souborového systému.

Tento způsob spočívá v ukládání médií jako souborů místo převodu do byte-verze souboru a následnému uložení do databáze, jež by s největší pravděpodobností vedlo k zahlcení a přetížení databáze jako je PostgreSQL. Databáze se v případě tohoto projektu využívá pouze k ukládání informací o souborech, jimiž jsou například typ a cesta odkazující na chat, kterému patří.

### 4.4 Vytváření událostí

Poslední stěžejní funkcí projektu je tvorba akcí. Oproti ostatním programům lze v této verzi akce volit nabízené termíny konání. Při tvorbě události autor zvolí několik různých návrhů termínů konání, na základě čehož může každý z pozvaných uživatelů hlasovat o svých časových preferencích.

K jednotlivým termínům existují 4 typy voleb. První z nich je akce nehlasování, tedy uživatel nezvolil nebo nechce o daném termínu hlasovat. Druhým typem je „preference“ některého z termínů. Dalším

typem je označení termínu jako „nevhodný“. A poslední možností je volba „pokud bude třeba“, jež by měl uživatel zvolit, pokud si pro tento termín v krajním případě udělá čas.

Vyhodnocení hlasování proběhne automaticky při dosažení časového data pro ukončení hlasování, které je taktéž zvoleno při tvorbě dané události. Jako vhodný termín se vybere ten s největším počtem hlasů pro tento termín.

## 4.5 Databáze

Databáze patří k jedné z nejpodstatnějších částí jakékoli webové aplikace. Ukládá nezbytná data k vykreslení obsahu a tkví v ní rychlost k dotazování těchto dat. Její struktura je uložena v souboru *db\_init.sql*, kde jsou příkazy k vytvoření všech potřebných tabulek tvořící nutné relace.

Pro zápis nebo přístup k těmto datům projekt využívá knihovnu *pg*. Ta umožňuje vykonávat jak samostatné příkazy, tak i pracovat v rámci takzvaného *poolu*. V projektu byl použit druhý způsob implementace, který je vhodný pro práci více uživatelů zároveň. Do databáze se tak přistupuje na bázi klientů.

Protože aplikaci používá v některých momentech vícero uživatelů zároveň, bylo nutné zajistit, aby si navzájem nepřepisovali svá data, ke kterým chtějí přistupovat ve stejném čase. Prostým řešením je implementace transakcí, které si s tímto problémem poradí. I díky tomuto byla zvolena v projektu právě databáze PostgreSQL, jež tyto funkce efektivně zvládá.

## 5. Návod k instalaci

V této kapitole se nachází návod k instalaci aplikace. Pojednává o zprovoznění databáze, tedy o tabulkách, které je třeba před použitím inicializovat, a jiných úkonech, jež je nutné ke spuštění aplikace provést.

V první řadě je třeba naklonovat repositář tohoto projektu z verzovací služby Github, který obsahuje všechny soubory projektu mimo soubor *.env.local*. Tento skrytý soubor obsahuje tajné parametry projektu:

```
-----  
DB_USERNAME=  
DB_PASSWORD=  
DB_HOST=localhost  
DB_PORT=5432  
DB_NAME=postgres  
LOGIN_EXPIRATION=1800000  
-----
```

Tyto údaje jsou třeba ke správnému fungování aplikace. Prvních pět definuje *PostgreSQL* databázi, jež budeme v projektu používat. Pole *LOGIN\_EXPIRATION* říká časový údaj v milisekundách určují dobu, po kterou má být každý uživatel po získání přihlašovacího tokenu autorizován. V základu je doba nastavená na 30 minut. Lze však zvolit i jiný časový údaj.

Další neméně důležitou částí je zprovoznění databázového serveru *PostgreSQL* dle návodu této technologie. Po úspěšném spuštění databáze je nutné inicializovat všechny tabulky, jež projekt využívá. Jejich předpis se nachází v souboru *./src/database/db\_init.sql* ve složce projektu.

Pro samotné spuštění je třeba nainstalovat správce balíčků pro Javascript, jež se nazývá *npm*. Díky němu se projekt spouští. Pokud je nainstalován, pak nezbyvá, než projekt zbuildovat pomocí příkazu v souboru *package.json*, tedy napsat příkaz *npm next build*. Ještě před samotným spuštěním je nutné spustit *server socket.io*,



který běží na portu 3001. Tento server lze spustit příkazem *npm next node*. Pak už zbývá jen poslední fáze, a to konkrétně spustit projekt na portu 3000 pomocí příkazu *npm next start*.

Všechny příkazy pro ovládání Next.js projektu společně s verzemi a názvy použitých knihoven se nachází v souboru *package.json*, který se nachází v kořenové složce projektu.

## 6. Závěr

Po mnoha úskalích projekt splnil svá hlavní očekávání, tedy podporuje funkce chatování a vytváření událostí s možností volby termínů. Je tedy schopna posloužit jakožto dobrý komunikační kanál pro skautská střediska.

Projekt však obsahuje též několik nedostatků, jež by bylo dobré implementovat. Ty se týkají například nahrávání videí, neboť aplikace podporuje jen fotky. Případně pak administrace středisek a oddílů, která v aktuální verzi není k dispozici.

Po závěrečném zhodnocení bych si tedy troufl tvrdit, že se projekt stal velice dobrým základem pro komplexnější sociální síť, která by mohla sloužit jako podpůrný nástroj k fungování skautských středisek nejen v Čechách, ale i ve zbytku světa.

## 7. Bibliografie

- [1] Vercel, „Next.js,“ Únor 2024. [Online]. Available: <https://nextjs.org/docs>.
- [2] Microsoft, „Dokumentace jazyka Typescript,“ leden 2024. [Online]. Available: <https://www.typescriptlang.org/docs/>.
- [3] Google, „Dokumentace Material UI,“ Prosinec 2023. [Online]. Available: <https://mui.com/material-ui/getting-started/>.
- [4] Meta, „Dokumentace React.js,“ Leden 2024. [Online]. Available: <https://react.dev/>.
- [5] T. P. G. D. Group, „Dokumentace relační databáze PostgreSQL,“ Listopad 2023. [Online]. Available: <https://www.postgresql.org/docs/>.
- [6] Socket.IO, „Dokumentace knihovny socket.io,“ Březen 2024. [Online]. Available: <https://socket.io/docs/v4/>.
- [7] „Dokumentace knihovny zod,“ Únor 2024. [Online]. Available: <https://zod.dev/>.
- [8] „Reozitář crypto.js,“ Leden 2024. [Online]. Available: <https://github.com/brix/crypto-js>.
- [9] „Dokumentace knihovny axios,“ Listopad 2023. [Online]. Available: <https://axios-http.com/docs/intro>.
- [10] „Návod k instalaci Material UI,“ Listopad 2023. [Online]. Available: <https://medium.com/frontendweb/how-to-install-material-ui-in-the-next-js-with-the-app-router-10d2a60e41f2#2959>.
- [11] Geeksforgeeks, „Solení a hashování,“ Leden 2024. [Online]. Available: <https://www.geeksforgeeks.org/what-is-salted-password-hashing/>.
- [12] „Návod k real-time chatovací webové aplikaci,“ Březen 2024. [Online]. Available: <https://blog.stackademic.com/building-a-real-time-chat-app-with-next-js-socket-io-and-typescript-e60ba40c09c7>.
- [13] „Knihovna Socket.io,“ [Online]. Available: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fsocket.io%2Fdocs%2Fv3%2Frooms%2F&psig=AOvVaw1elwUMXn6tPSRfy33ZU5SP&ust=1711290899062000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCMCL3sPNioUDFQAAAAAdAAAAABAE>. [Přístup získán 2024].