



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**WEBOVÁ KNIHOVNA GRAFICKÝCH
KOMPONENT PRO VIZUALIZACI DAT
Z HLASOVÁNÍ ZASTUPITELSTEV**

WEB-BASED LIBRARY OF GRAPHICAL COMPONENTS FOR COUNCIL VOTING DATA
VISUALISATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARCIN SOCHACKI

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JIŘÍ HYNEK, Ph.D.

BRNO 2025

Zadání bakalářské práce



164730

Ústav: Ústav informačních systémů (UIFS)
Student: **Sochacki Marcin**
Program: Informační technologie
Název: **Webová knihovna grafických komponent pro vizualizaci dat z hlasování zastupitelstev**
Kategorie: Webové aplikace
Akademický rok: 2024/25

Zadání:

1. Seznamte se s problematikou zpracování dat a vizualizace hlasování samospráv (města, kraje, apod.). Prostudujte typy analýz na těchto datech.
2. Prostudujte principy a techniky vizualizace dat, typy vizualizačních nástrojů a dostupných technologií (např. D3.js).
3. Analyzujte dostupná otevřená data z hlasování samospráv. Vyhodnoťte požadavky na odvozování znalostí z těchto dat.
4. Navrhněte sadu vizualizačních komponent určených pro odvozování znalostí z dat analyzovaných v bodě 3.
5. Po konzultaci s vedoucím navržené komponenty implementujte.
6. Otestujte funkčnost a použitelnost implementovaných komponent na datech z vybraných samospráv.

Literatura:

- Few, S. (2006). Information Dashboard Design. (n.p.): O'Reilly Media, Incorporated.
- Murray, S. (2017). Interactive Data Visualization for the Web: An Introduction to Designing with D3. O'Reilly Media.
- Schwabish, J. (2021). Better Data Visualizations: A Guide for Scholars, Researchers, and Wonks. Columbia University Press.
- Tufte, E. R. (2001). The Visual Display of Quantitative Information. Graphics Press.
- Zaklová, K. (2023). Analýza a vizualizace dat z hlasování Zastupitelstva města Brna. Diplomová práce. Vedoucí práce: Ing. Jiří Hynek Ph.D. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/146480>

Při obhajobě semestrální části projektu je požadováno:
Body 1 - 4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hynek Jiří, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1.11.2024
Termín pro odevzdání: 14.5.2025
Datum schválení: 22.10.2024

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací webové knihovny vizualizačních komponent určených pro vizualizaci dat z hlasování zastupitelstev měst a krajů. Komponenty byly navrženy tak, aby byly univerzálně použitelné, snadno konfigurovatelné pomocí parametrů a aby podporovaly funkcionální přístup při práci s daty. Knihovna byla integrována do systému Zastupko.cz, kde byly komponenty testovány na reálných datech z krajů. Výsledkem je sada flexibilních komponent, které vývojářům usnadní práci s daty, zlepší srozumitelnost výstupů a přispějí k vyšší transparentnosti rozhodování v samosprávách.

Abstract

This bachelor's thesis focuses on the design and implementation of a web-based library of visualization components for displaying data from municipal and regional council voting. The components are designed to be universally reusable, easily configurable through parameters, and to support a functional approach to data handling. The library was integrated into the Zastupko.cz system, where the components were tested on real-world data from regions. The result is a set of flexible components that make it easier for developers to work with data, improve the clarity of visual outputs, and contribute to greater transparency in local government decision-making.

Klíčová slova

vizualizace dat, hlasování zastupitelstev, webové komponenty, otevřená data, React, Chart.js, D3.js

Keywords

data visualization, council voting, web components, open data, React, Chart.js, D3.js

Citace

SOCHACKI, Marcin. *Webová knihovna grafických komponent pro vizualizaci dat z hlasování zastupitelstev*. Brno, 2025. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jiří Hynek, Ph.D.

Webová knihovna grafických komponent pro vizualizaci dat z hlasování zastupitelstev

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Hynka, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Marcin Sochacki

12. května 2025

Poděkování

Tímto bych rád poděkoval Ing. Jiřímu Hynkovi, Ph.D. za vedení této bakalářské práce, za jeho čas, rady a pravidelné konzultace během celého řešení.

Obsah

1	Úvod	4
2	Zpracování dat a vizualizace hlasování samospráv	5
2.1	Otevřená data samospráv a jejich význam	5
2.1.1	Zastupitelstva a jejich složení	6
2.1.2	Zveřejňování otevřených dat městy	6
2.1.3	Co jsou otevřená data a pětihvězdičkové schéma	6
2.2	Existující aplikace	7
2.2.1	HowTheyVote.eu	7
2.2.2	Otvorená samospráva	8
2.2.3	Zastupko.cz	9
3	Principy a techniky vizualizace dat	11
3.1	Přínosy vizualizace dat	11
3.2	Historie vizualizace dat	11
3.3	Efektivní vizualizace a její principy	12
3.4	Techniky vizualizace dat	12
3.4.1	Statická vizualizace dat	12
3.4.2	Interaktivní vizualizace dat	13
3.4.3	Porovnání obou přístupů	13
3.5	Knihovny pro programátory	14
3.5.1	Práce s grafickými prvky na nízké úrovni	14
3.5.2	Nástroje pro řízení vizualizací nad SVG/Canvas	15
3.5.3	Vizuální knihovny s hotovými komponentami	16
4	Analýza problému	18
4.1	Analýza dat	18
4.1.1	Data z městských zastupitelstev	18
4.1.2	Krajská zastupitelstva	19
4.2	Analýza aktuálních vizualizací v projektu Zastupko.cz	19
4.2.1	Mandáty	19
4.2.2	Docházka zastupitelů	20
4.2.3	Porovnání zastupitelů	21
4.2.4	Nejednotnost stran	21
4.2.5	Mapové vizualizace	22
4.3	Nové požadavky na vizualizace	23
5	Návrh nových vizualizačních komponent	24

5.1	Vizualizace mandátů	24
5.2	Časová osa aktivit	25
5.3	Sloupcový graf	26
5.4	Čárový graf	27
5.5	Kombinovaný graf	28
5.6	Vizualizace s podporou období	29
5.7	Mapová vizualizace	30
6	Implementace	31
6.1	Vizualizace mandátů	31
6.2	Časová osa aktivit	32
6.3	Sloupcový graf	32
6.4	Čárový graf	33
6.5	Kombinovaný graf	34
6.6	Vizualizace s podporou období	35
6.7	Mapová vizualizace	36
7	Testování	37
7.1	Lokální testování	37
7.2	Testování v projektu Zastupko.cz	37
7.3	Praktické použití vizualizací	38
8	Závěr	40
	Literatura	41

Seznam obrázků

2.1	Příklad vizualizace hlasování na platformě HowTheyVote.eu – hlasování o situaci v Gruzii.	8
2.2	Ukázka hodnocení města Bratislava – Staré Mesto	9
2.3	Vizualizace nejednotnosti stran na Zastupko.cz	10
3.1	Základní části grafu v Highcharts [15].	17
4.1	Rozložení mandátů v Zastupitelstvu města Brna (2022–2026)	20
4.2	Rozložení mandátů v Radě města Mostu (2022–2026)	20
4.3	Vizualizace docházky zastupitelů a průměrné docházky podle stran	21
4.4	Shoda v hlasování mezi dvěma vybranými zastupiteli podle jednotlivých zasedání	21
4.5	Jak moc jsou strany jednotné při hlasování	22
4.6	Mapová vizualizace odměn zastupitelů podle krajů	22
5.1	Návrh půlkruhového zobrazení mandátů v zastupitelstvu	25
5.2	Návrh vzhledu časové osy aktivit	26
5.3	Ukázka návrhu režimu s průhledností sloupcového grafu	27
5.4	Ukázka čárového grafu s barevným vyplněním pod čarou	28
5.5	Ukázka kombinovaného grafu	28
5.6	Ukázka grafu s barevně vyznačenými obdobími	29
6.1	Půlkruhová vizualizace mandátů v Chart.js	31
6.2	Výsledné zobrazení sloupcového grafu	33
6.3	Ukázka různých variant čárového grafu podle nastavení parametrů	34
6.4	Ukázka horizontální varianty kombinovaného grafu	35
7.1	StackedBar vizualizace hlasování podle volebních období [27]	38
7.2	Čárový graf vývoje souhlasnosti jednotlivých poslaneckých stran [27]	39
7.3	Příklad čárového grafu z práce zaměřené na parlamentní hlasování [17]	39

Kapitola 1

Úvod

Transparentnost místní správy je klíčová pro správné fungování demokracie. Lidé mají mít možnost sledovat, jak jejich zvolení zástupci rozhodují o důležitých tématech. Způsob, jakým zastupitelé hlasují, může veřejnosti nabídnout důležitý pohled na jejich práci i na samotný rozhodovací proces. Bohužel se v praxi často stává, že tato data nejsou snadno k nalezení nebo nejsou prezentována způsobem, který by byl běžným lidem dobře srozumitelný. To pak vede k tomu, že veřejnost, odborníci i novináři mají problém tato data využít k hlubší analýze a k lepšímu informování společnosti.

Existují nástroje, jako například *Zastupko.cz*¹, které se snaží tuto situaci zlepšit a poskytují data o hlasování zastupitelů v přehlednější podobě. Cílem této práce je zlepšit přístup k těmto datům a umožnit jejich lepší porozumění a využití. Mé výstupy budou přímo zaměřeny na zlepšení vizualizace dat pro *Zastupko.cz*, což umožní lidem lépe chápat rozhodovací procesy, analýzy a přispět k informovanějšímu rozhodování. Tato práce má za úkol připravit konkrétní nástroje pro efektivní vizualizaci dat, které zvýší jejich přehlednost, a tím podpoří širší veřejnost i odborníky v jejich využívání pro analýzu.

I když existují grafické knihovny pro vizualizaci dat, moje práce se zaměřuje na přizpůsobení těchto knihoven konkrétním datům o hlasování zastupitelů. Mým cílem je vzít tyto existující knihovny a upravit je tak, aby byla data o samosprávách snadno a přehledně vizualizována. Cílem je tyto knihovny přizpůsobit tak, aby umožnily jednoduché, přehledné a přizpůsobitelné vizualizace pro specifická data z hlasování v zastupitelstvech měst a krajů.

Tato práce se zaměří na vytvoření webové knihovny grafických komponent pro vizualizaci těchto dat, která bude připravena na konkrétní data samospráv. Bude to knihovna, která poskytne nástroje pro efektivní prezentaci dat o hlasování, a bude navržena tak, aby ji bylo možné snadno přizpůsobit různým typům hlasování a typům zastupitelstev. Bude kladen důraz na jednoduchost použití a přehlednost výstupů, což znamená, že výsledné vizualizace budou snadno čitelné a použitelné jak pro odborníky, tak pro veřejnost. Tímto způsobem se podaří zvýšit porozumění těmto datům a usnadnit jejich analýzu.

Kapitola č. 2 představuje problematiku otevřených dat a existující aplikace. V kapitole č. 3 jsou popsány vhodné vizualizační techniky a existující knihovny. Kapitola č. 4 se věnuje analýze aktuálního stavu a definici požadavků. Následuje návrh komponent v kapitole č. 5 a jejich realizace v kapitole č. 6. Testování a praktické využití vizualizací je shrnuto v kapitole č. 7. Závěrem kapitola č. 8 hodnotí výsledek práce.

¹<https://zastupko.cz/>

Kapitola 2

Zpracování dat a vizualizace hlasování samospráv

Podle studie [2] existují tři hlavní důvody pro zveřejňování vládních dat, které lze dobře aplikovat i na data o hlasování v samosprávách:

- **Transparentnost** je klíčová pro fungování demokratické společnosti. Občané potřebují přístup k datům, aby mohli sledovat činnost vlády nebo zastupitelů a posuzovat jejich legitimitu. Transparentnost zahrnuje nejen samotný přístup k datům, ale i možnost jejich dalšího zpracování, sdílení a využití. Lepší přehled o fungování veřejné správy také přispívá k většímu zapojení občanů a lepší veřejné kontrole [2].
- **Uvolnění sociální a komerční hodnoty dat** znamená, že data shromažďovaná státem mohou najít uplatnění v různých oblastech, například ve školství, životním prostředí nebo při plánování rozvoje. Jejich zveřejnění podporuje inovace a umožňuje vznik nových aplikací a služeb [2].
- **Podpora participativního vládnutí** se týká zapojení občanů do rozhodovacích procesů. Otevřená data dávají veřejnosti možnost lépe se orientovat v činnosti samosprávy a aktivně reagovat na její kroky. To platí i pro data o hlasování, která zvyšují informovanost veřejnosti a podporují účast na demokratickém dění nejen během voleb [2].

Podle studie [14] se v posledních letech objevuje trend využívat technologie jako umělá inteligence (AI) a internet věcí (IoT) ke zlepšení práce s otevřenými vládními daty. Například díky umělé inteligenci je možné z otevřených dat snadněji zjistit zajímavé informace, kterých by si člověk sám nevšiml. Některé nástroje dokážou automaticky najít chyby nebo neobvyklé situace v datech. Díky tomu mohou vznikat takzvané „chytré“ systémy, které pomáhají reagovat rychleji a lépe využívat data pro veřejnost.

Data o hlasování jsou důležitým prostředkem pro hodnocení kvality veřejné správy a jejich zveřejňování zvyšuje zodpovědnost zvolených zástupců, čímž přispívá k větší otevřenosti a efektivnějšímu rozhodovacímu procesu [2].

2.1 Otevřená data samospráv a jejich význam

Data o hlasování v samosprávách obsahují klíčové informace o rozhodovacích procesech v městských a krajských samosprávách. Tyto informace umožňují občanům i odborníkům

lépe pochopit, jak volení zástupci vykonávají své funkce a jaké mají názory. V České republice je zveřejňování těchto dat podporováno legislativou, konkrétně zákonem č. 106/1999 Sb., o svobodném přístupu k informacím [32]. Tento zákon upravuje pravidla pro přístup k informacím veřejné správy a dává občanům právo získat přehled o činnosti veřejných institucí, včetně výsledků hlasování zastupitelstev.

2.1.1 Zastupitelstva a jejich složení

Zastupitelstvo je nejvyšším samosprávným orgánem obce. Společně s dalšími orgány, jako jsou rada obce, starosta nebo obecní úřad, zajišťuje řízení obce. Každá obec, město nebo městys má své zastupitelstvo, jehož počet členů se odvíjí od počtu obyvatel a velikosti obce. Je tvořeno volenými členy, kteří získávají mandát prostřednictvím voleb. Mandát vzniká okamžikem ukončení hlasování a je podmíněn složením zákonného slibu. Členové zastupitelstva vykonávají svou funkci osobně, v souladu se zákonem a bez jakýchkoli příkazů. Zasedání zastupitelstva se koná nejméně jednou za tři měsíce a svolává je starosta. Program zasedání připravuje rada obce nebo starosta, přičemž návrhy na body programu mohou podávat i jednotliví členové zastupitelstva nebo výbory. Průběh zasedání řídí starosta a musí být z něj pořízen zápis. Pro přijetí rozhodnutí je třeba, aby byla přítomna nadpoloviční většina členů [22, 33].

2.1.2 Zveřejňování otevřených dat městy

Zpráva Ministerstva pro místní rozvoj a Ministerstva vnitra z roku 2017 [21] ukazuje, že způsob, jakým města v České republice zveřejňují otevřená data, se liší. Některá města data zveřejňují podle platné legislativy a metodik, jiná je poskytují jen částečně, nebo je vůbec neotevívají. Podle šetření:

- pouze 15 % měst (6 z 40) zveřejňuje otevřená data v souladu se zákonem č. 106/1999 Sb. a standardy Ministerstva vnitra.
- dalších 22,5 % měst poskytuje data ve strojově čitelném formátu, ale bez splnění všech požadavků, například nejsou katalogizována v Národním katalogu otevřených dat.
- zbytek měst data buď teprve plánuje zveřejnit, nebo je vůbec nezveřejňuje.

Zpráva také uvádí příklady dobré praxe, jako je město Ostrava, které poskytuje otevřená data včetně geografických informací a dat ze svých institucí. Ostrava zároveň dodržuje metodiky publikace a katalogizace otevřených dat stanovené Ministerstvem vnitra [21].

2.1.3 Co jsou otevřená data a pětihvězdičkové schéma

Otevřená data jsou informace, které jsou dostupné na internetu ve formátu, který umožňuje jejich volné použití, sdílení a další zpracování bez nutnosti žádat o svolení nebo platit poplatky. Cílem otevřených dat je umožnit jejich opětovné využití veřejností, akademiky, vývojáři nebo soukromými firmami [25].

Z hlediska kvality publikovaných dat lze využít takzvané schéma 5 hvězdiček otevřených dat, které v roce 2010 navrhl tvůrce webu Sir Tim Berners-Lee. Toto schéma slouží jako jednoduchý návod pro poskytovatele dat, jak postupně zvyšovat úroveň otevřenosti a využitelnosti jejich dat. Každá hvězdička představuje krok směrem k větší strojové čitelnosti,

Počet hvězdiček	Popis
★	Data jsou dostupná na webu s otevřenou licencí, bez ohledu na formát.
★★	Data jsou dostupná ve strojově čitelném formátu.
★★★	Data jsou ve formátu, který nevyžaduje proprietární software – např. CSV.
★★★★	Data využívají otevřené standardy jako RDF nebo SPARQL (standardy W3C).
★★★★★	Data jsou propojena s jinými daty – např. přes odkazy na DBpedia, čímž se zvyšuje jejich kontextová hodnota.

Tabulka 2.1: Pětihvězdičkové schéma otevřených dat podle Tima Berners-Leeho [26]

otevřenosti a propojitelnosti dat na webu (tzv. Linked Open Data). Přehled jednotlivých úrovní poskytuje tabulka 2.1.

Abychom vůbec mohli mluvit o datech, je vhodné rozlišit několik úrovní informačních variant. Data jsou surové, zaznamenané hodnoty, například čísla nebo texty, které samy o sobě nemusí mít význam. Teprve když je člověk nebo počítač na základě znalostí interpretuje, stávají se z nich informace. Znalosti pak vznikají jako schopnost člověka nebo stroje tyto informace chápat v širších souvislostech a využívat je při rozhodování [1].

2.2 Existující aplikace

Aby se zlepšila přístupnost a transparentnost dat o hlasování, vzniklo pár online platform zaměřených na prezentaci těchto informací. Tyto aplikace pomáhají občanům lépe sledovat rozhodování, analyzovat postoje politiků a získat podrobné informace o fungování samospráv nebo jiných veřejných institucí. Díky vizualizačním nástrojům a interaktivním funkcím jsou data přehlednější a uživatelé mají lepší vhled na danou situaci. V další části budou představeny některé aplikace, které podporují transparentnost a vizualizaci dat v rámci samospráv i na evropské úrovni.

V této kapitole jsou představeny tři konkrétní platformy, které reprezentují různé úrovně veřejné správy: *HowTheyVote.eu* na úrovni Evropského parlamentu, *Otvorená samospráva* zaměřená na slovenská města a kraje a *Zastupko.cz*, které se specializuje na české městské a krajské zastupitelstva. Vybral jsem tyto tři platformy, protože každá z nich ukazuje jiný přístup k prezentaci hlasovacích dat, jedna na evropské úrovni, druhá na Slovensku a třetí v České republice.

2.2.1 HowTheyVote.eu

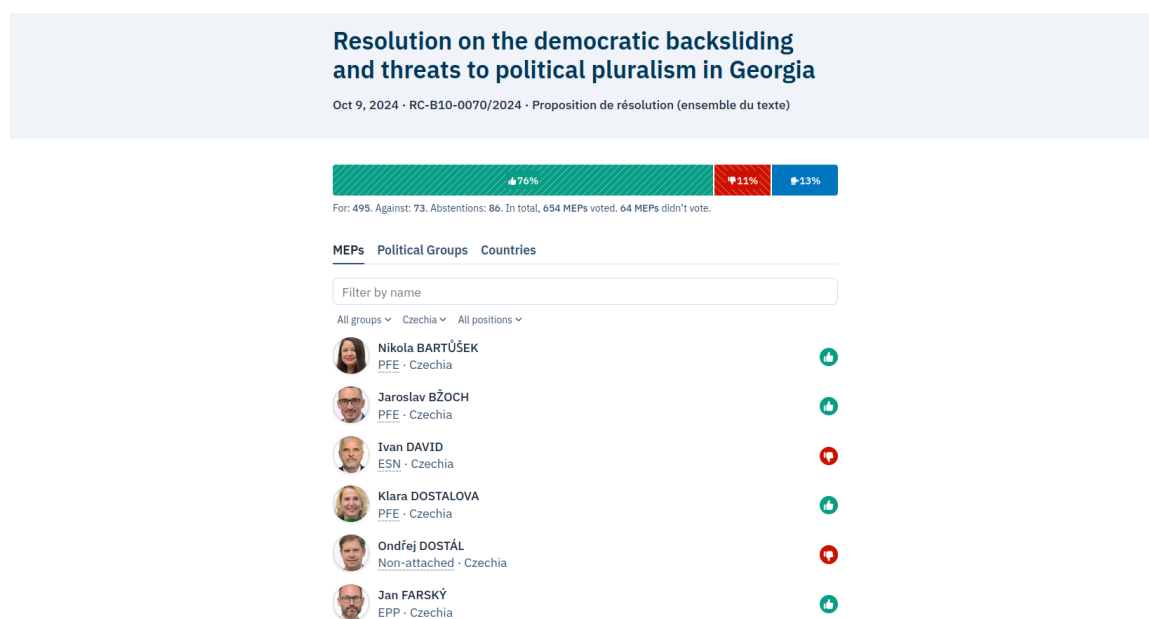
*HowTheyVote.eu*¹ je online platforma, která zpřístupňuje informace o hlasování členů Evropského parlamentu. Uživatelé zde mohou najít detailní informace o jednotlivých hlasováních, včetně názvu, data a počtu hlasů *pro*, *proti* a *zdržel se*. Tímto způsobem platforma poskytuje rychlý přehled o výsledcích hlasování a míře podpory či opozice mezi europoslanci.

Platforma umožňuje filtrování výsledků podle různých kritérií, například podle země, politické frakce nebo konkrétního europoslance. Díky vyhledávací funkci můžou uživatelé

¹<https://howtheyvote.eu>

snadno nalézt konkrétní jména a sledovat práci zvolených zástupců. Tohle umožňuje analyzovat postoje poslanců z různých států a politických frakcí. Další užitečnou funkcí je přehledný seznam poslanců, který obsahuje informace o jejich politické příslušnosti a funkcích v Evropském parlamentu. Toto umožňuje sledovat konzistenci jejich hlasování a porovnat ji s jejich programy a postoji.

Vizualizace hlasování jsou na platformě prezentovány pomocí různých grafů, jako jsou barevné pruhy zobrazující podíl hlasů *pro*, *proti* a *zdržel se*. Příkladem může být vizualizace na obrázku 2.1, která ukazuje hlasování o usnesení týkajícím se politické situace v Gruzii. Na pruhu nahoře lze přehledně vidět celkový poměr hlasů, zatímco seznam níže ukazuje, jak konkrétní europoslanci zvolené země (v tomto případě z Česka) hlasovali.



Obrázek 2.1: Příklad vizualizace hlasování na platformě HowTheyVote.eu – hlasování o situaci v Gruzii.

Platforma také umožňuje sledovat hlasování podle témat, jako jsou klima, ekonomika nebo lidská práva. Uživatelé tak mají lepší přehled o tom, jak se různí poslanci staví k otázkám, které jsou pro občany důležité. Tato tematická kategorizace pomáhá snadněji vyhledávat a rychle se dostat k potřebným informacím.

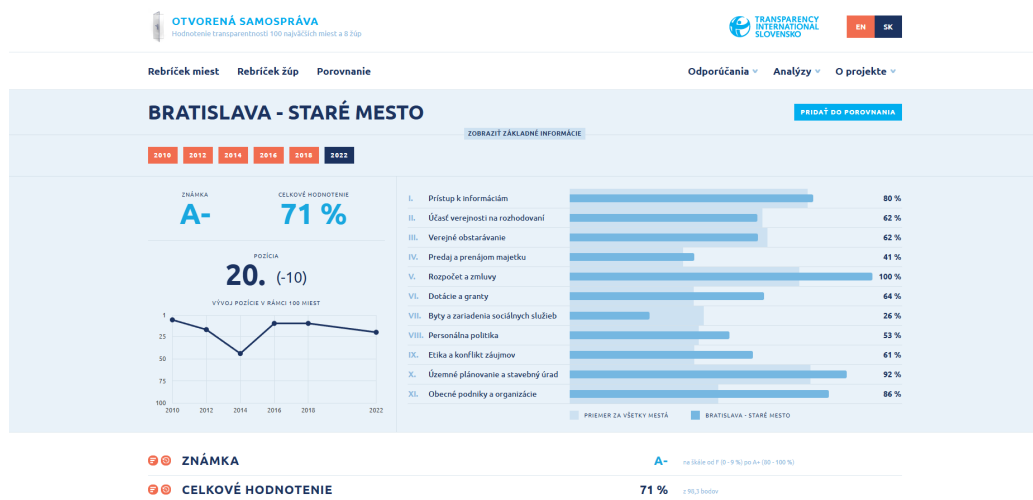
Design platformy je moderní a uživatelsky přívětivý, optimalizovaný pro různé typy zařízení. Navigace je snadná a intuitivní, což umožňuje rychlé nalezení potřebných informací. Stránka je vhodná jak pro běžné občany, tak pro odborníky, kteří chtějí sledovat práci svých zástupců.

2.2.2 Otvorená samospráva

Projekt *Otvorená samospráva*² od Transparency International Slovensko se zaměřuje na to, jak transparentní jsou slovenská města a kraje. Nabízí nástroje, které pomáhají sledovat a porovnávat otevřenost veřejné správy. Hodnocení pro jednotlivá města nebo kraje se zobrazuje v procentech a známkách, například „A-“ pro Bratislavu – Staré Mesto jak je

²<https://samosprava.transparency.sk/>

zobrazeno na obrázku 2.2. Toto skóre vychází z několika různých kritérií, které pokrývají různé oblasti transparentnosti. Součástí každého hodnocení je i graf, který ukazuje, jak se skóre vyvíjelo v průběhu let, co pomáhá sledovat změny.



Obrázek 2.2: Ukázka hodnocení města Bratislava – Staré Mesto

Platforma obsahuje vizualizace, které porovnávají výsledky celkového hodnocení transparentnosti daného města nebo kraje s průměrem všech hodnocených samospráv. Vizualizace mají formu sloupcových grafů a pokrývají několik důležitých oblastí:

- **Přístup k informacím:** Hodnotí jak snadno mají občané přístup k veřejným informacím.
- **Účast veřejnosti na rozhodování:** Sleduje zapojení občanů do rozhodovacích procesů měst a krajů.
- **Veřejné obstarávání:** Zobrazuje transparentnost při výběrových řízeních a veřejných zakázkách.
- **Dotace a granty:** Sleduje, jak transparentně jsou přidělovány veřejné finanční prostředky.

Tyto vizualizace poskytují přehled o tom, kde se města či kraje nacházejí v konkrétních oblastech. Uživatelé mohou rychle zjistit, v čem jejich obec vyniká a na čem je třeba pracovat. Kromě samotného hodnocení a vizualizací nabízí platforma i možnost porovnávání jednotlivých měst a krajů. Tato funkce umožňuje uživatelům srovnávat, jak si různé regiony vedou v oblasti transparentnosti, a analyzovat rozdíly mezi nimi. Ačkoliv tento projekt nezahrnuje konkrétní data o hlasování zastupitelstev, rozhodl jsem se ho do práce zařadit, protože ukazuje jiný přístup ke sledování fungování samospráv. Zároveň ukazuje, jak jde vizualizovat další aspekty transparentnosti veřejné správy.

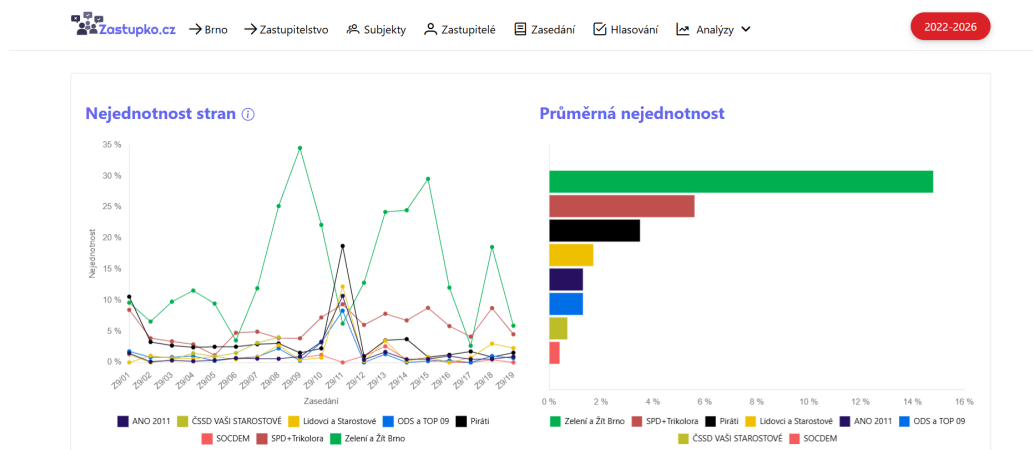
2.2.3 Zastupko.cz

*Zastupko.cz*³ je webová stránka, která zlepšuje přístup k informacím o hlasování městských a krajských zastupitelstev. Projekt vznikl na základě diplomové práce [30] a postupně se

³<https://zastupko.cz/>

rozšířil o další funkcionalitu. Uživatelé zde najdou přehled o zastupitelích, zasedáních a výsledcích hlasování.

Platforma nabízí různé vizualizace, které usnadňují orientaci v datech a jejich analýzu. Jedním z prvků je například graf na obrázku 2.3 zobrazující nejednotnost stran, který ukazuje, jak moc se jednotliví členové stran liší ve svých hlasováních. Příklad takového grafu pro město Brno znázorňuje, jak se nejednotnost měnila během zasedání, a průměrnou míru nesouladu u každé strany. To pomáhá rychle zjistit, které strany drží pevně své názory a kde jsou časté rozdíly v postojích.



Obrázek 2.3: Vizualizace nejednotnosti stran na Zastupko.cz

Jde tam najít grafy, které ukazují, jak hlasovali jednotliví zastupitelé. Uživatelé si tak mohou zjistit, jaký postoj měli k různým tématům a jak se jejich rozhodnutí shodovala nebo lišila od ostatních zastupitelů. Zastupko také umožňuje filtrovat data podle různých kritérií, například podle politických stran, období nebo konkrétních hlasování. Uživatelé si mohou prohlédnout jednotlivá zasedání a zjistit, jak strany nebo zastupitelé hlasovali. Tato funkce poskytuje rychlý přístup k potřebným informacím a je užitečná pro každého, kdo se zajímá o konkrétní témata nebo rozhodnutí.

Díky těmto funkcím zvyšuje Zastupko.cz transparentnost veřejné správy a pomáhá občanům lépe porozumět práci jejich zastupitelů. S vizualizacemi a snadným ovládáním mohou uživatelé lépe sledovat, jak jsou jejich zájmy zastupovány a jaké postoje mají jejich zástupci.

Kapitola 3

Principy a techniky vizualizace dat

Vizualizace dat představuje vizuální zobrazení dat prostřednictvím grafických prvků, jako jsou grafy nebo animace [20]. Podle Tamary Munzner lze vizualizaci definovat jako počítačově podporované vizuální zobrazení dat, jehož účelem je pomoci uživatelům efektivněji plnit úkoly. Cílem vizualizace je pomáhat lidem lépe porozumět datům. Design vizualizací zahrnuje mnoho kompromisů, od volby vhodného způsobu zobrazení až po interakci s uživatelem [24]. Stephen Few ve své knize *Information Dashboard Design* [12] definuje vizualizaci ve formě nástrojů zvaných *dashboard* jako vizuální zobrazení nejdůležitějších informací, které jsou uspořádány na jedné obrazovce tak, aby mohly být snadno monitorovány na první pohled. Takové zobrazení kombinuje grafické a textové prvky a zaměřuje se na efektivní komunikaci informací. Nejdůležitější na dashboardech je jejich schopnost poskytnout rychlý přehled a upozornit uživatele na důležité informace.

3.1 Přínosy vizualizace dat

Vizualizace dat umožňuje rychlé a snadné pochopení složitých informací tím, že je převádí do přehledné vizuální podoby. Pomáhá zvýraznit podstatné prvky, odstranit šum a umožňuje odhalit vzory i složité vztahy v datech. Lidský mozek přirozeně reaguje na barvy a tvary, což zvyšuje efektivitu při rozhodování. Vizualizace se tak stává nezbytným nástrojem pro datovou vědu, analytiku i každodenní rozhodování, a to napříč obory od financí až po medicínu [20]. Další přínos vizualizace spočívá v jejím využití jako externí reprezentace. Podle Tamary Munzner [24] vizualizace pomáhá uživatelům snížit kognitivní zátěž tím, že část zpracování informací přenáší na vizuální systém. Díky tomu je snazší rozpoznat důležité vzory a lépe propojit informace do jednoho kontextu. Diagramy a grafy, pečlivě navržené podle potřeb uživatele, mohou hodně urychlit proces hledání a rozpoznávání klíčových informací v datech. Zajímavý pohled na účel vizualizace nabízí také studie *What is visualization really for?* [9], která zdůrazňuje šetření času jako hlavní přínos vizualizace.

3.2 Historie vizualizace dat

Jak uvádí Stephen Few ve své knize *Information Dashboard Design* [12], historie vizualizace dat se začala formovat již v 80. letech s nástupem Exekutivních informačních systémů (EIS), které byly navrženy tak, aby zobrazovaly klíčové finanční ukazatele jednoduchým způsobem. Tato první implementace vizualizací však zůstala omezená pro vrcholové manažery a neměla širší využití. Hlavním problémem byla nepřipravenost technologií. Data byla

neúplná, nespolehlivá a rozptýlená mezi různými zdroji. Vize EIS tak na několik desetiletí upadla do pozadí, dokud nebyly technologie dostatečně vyspělé, aby to oživily [12].

V 90. letech přinesl rozvoj datových skladů, online analytického zpracování (OLAP) a business intelligence potřebné nástroje k integraci, zpracování a spolehlivému uchovávání dat. Tyto technologie vytvořily základy pro nový přístup k řízení založený na metrikách, zejména díky konceptu Balanced Scorecard, který představili Robert S. Kaplan a David P. Norton. Tento přístup, který se soustředí na klíčové ukazatele výkonnosti (KPIs), pomohl rozvoji vizualizačních nástrojů, které dnes tvoří základ moderního řízení výkonnosti firem (BPM).

Po skandálu společnosti Enron v roce 2001, který zvýšil tlak na korporace, aby prokázaly svou schopnost efektivně monitorovat výkon a zajistit odpovědnost vůči akcionářům. Tato situace vedla k prudkému rozvoji řešení typu *dashboard*, kdy si mnoho firem uvědomilo jejich potenciál jako nástrojů pro snadné sledování klíčových metrik. Trh brzy zaplavila široká nabídka softwarových řešení dashboardů, což podpořilo popularitu vizualizačních nástrojů, jak je známe dnes.

3.3 Efektivní vizualizace a její principy

Efektivní vizualizace dat je klíčová pro rychlé a přesné zprostředkování informací bez zbytečného zatížení uživatele. Willard C. Brinton již v roce 1923 napsal, že způsob prezentace dat může být často důležitější než samotná data, protože správně navržená vizualizace může výrazně ovlivnit jejich srozumitelnost a dopad [19].

Jedním ze základních principů je správný výběr vizualizačních prvků. Například sloupcové grafy jsou vhodné pro porovnávání kategorií, spojnicové grafy zobrazují trendy a změny v čase, zatímco mapy jsou ideální pro zodpovězení geografických otázek. Naopak špatně zvolený typ grafu může zkreslit informace a ztížit jejich pochopení [28].

Integrita vizualizace je dalším klíčovým aspektem. Bigwoodová a Sporeová definují efektivní graf jako takový, který je jednoduchý, jasný a přehledný [19]. Graf by měl být navržen tak, aby sděloval informace pravdivě a transparentně, bez zkreslení nebo manipulace [19]. Tamara Munzner upozorňuje, že návrh vizualizace je často složitý kvůli kompromisům mezi různými cíli, jako je estetika, přesnost a účelnost [24]. Je proto důležité navrhovat vizualizace, které respektují jak technické, tak lidské limity vnímání [24].

3.4 Techniky vizualizace dat

V praxi existují různé přístupy k tomu, jak zobrazovat data pomocí vizualizací. Nejčastěji se rozlišují dvě hlavní techniky: **statická vizualizace** a **interaktivní vizualizace**. Obě slouží ke stejnému účelu a to zobrazit data srozumitelným způsobem pomocí grafických prvků, ale liší se v tom, jakým způsobem s daty pracují a co nabízí uživateli [18].

3.4.1 Statická vizualizace dat

Statická vizualizace je jedním z nejběžnějších přístupů k zobrazování dat. Jedná se o pevně definované grafické výstupy, které jsou nezměnitelné a slouží k prezentaci jednoho konkrétního příběhu dat [5]. Typickým příkladem statických vizualizací jsou infografiky, které kombinují text, grafy a obrázky v přehledném rozložení [7]. Dalším příkladem mohou být statické grafy, jako jsou sloupcové a spojnicové grafy, často vytvářené pomocí nástrojů jako MS Excel [18].

Výhodou statických vizualizací je jejich jednoduchost, rychlost zpracování a nižší náklady. K jejich vytvoření nejsou zapotřebí pokročilé technologie ani školení uživatelů, což je činí snadno dostupnými i pro menší organizace nebo projekty [18]. Statické grafy jsou vhodné pro situace, kdy je třeba sdělit jasný a jednoznačný příběh bez potřeby další interakce [5].

Na druhou stranu statické vizualizace nepodporují hlubší průzkum dat. Uživatel nemůže měnit zobrazené hodnoty ani analyzovat data z různých úhlů pohledu. Kvůli těmto omezením nejsou statické vizualizace vhodné pro práci s velkými a složitými datovými sadami, které vyžadují hlubší analýzu [18].

3.4.2 Interaktivní vizualizace dat

Interaktivní vizualizace představují modernější přístup k prezentaci dat, který umožňuje uživatelům aktivně se zapojit do procesu analýzy. Tyto vizualizace jsou často realizovány prostřednictvím webových nebo softwarových aplikací, kde uživatelé mohou pomocí kliknutí myši, zadávání vstupů z klávesnice nebo jiných interakcí zkoumat data a odhalovat nové souvislosti [18]. Například interaktivní dashboardy, vytvořené ve specifických nástrojích, umožňují uživatelům přizpůsobit si zobrazení dat podle svých konkrétních potřeb [5].

Jedním z hlavních přínosů interaktivních vizualizací je jejich schopnost zpracovávat rozsáhlé a složité datové sady. Díky tomu jsou ideální pro situace, kde je potřeba analyzovat velké množství dat [18]. Interaktivní vizualizace se také zaměřují na čtenářsky řízený přístup, což znamená, že uživatel může aktivně ovlivňovat výsledky a přizpůsobovat je svým požadavkům [18].

Nevýhodou tohoto je složitost a náklady. K vytvoření interaktivních vizualizací je často zapotřebí speciální software, technické dovednosti. Uživatelé také mohou potřebovat školení, aby mohli efektivně pracovat s interaktivními nástroji [18]. Navíc interaktivní vizualizace mohou být časově náročnější na vývoj a implementaci, což zvyšuje jejich celkovou finanční náročnost [5].

3.4.3 Porovnání obou přístupů

Statická a interaktivní vizualizace představují dva hlavní přístupy k prezentaci dat, které se výrazně liší ve svých vlastnostech, výhodách i omezeních. Statická vizualizace je pevně daná a neměnná, zaměřená na jeden konkrétní příběh dat. Naproti tomu interaktivní vizualizace umožňuje uživatelům aktivně se zapojit do procesu analýzy a zkoumat data z různých perspektiv. Tato flexibilita je zvláště užitečná při práci s velkými a složitými datovými sadami [7, 5].

Statická vizualizace je vhodná pro rychlé a jednoduché prezentace, kde je třeba sdělit jasný a jednoznačný příběh. Uživatelé nemají možnost interagovat s daty nebo je dále analyzovat. Tento přístup je méně nákladný, rychlejší na přípravu a jednodušší na pochopení, což ho činí ideálním pro prezentace nebo tiskové materiály [18]. Interaktivní vizualizace naopak nabízí širší možnosti. Uživatelé mohou zadávat vstupy, prohlížet data z různých úhlů a hledat nové souvislosti. Tento přístup vyžaduje pokročilejší nástroje, technické dovednosti a často vyšší náklady na vývoj [5].

Rozdíly mezi těmito dvěma přístupy shrnuje následující tabulka 3.1, která ukazuje klíčové aspekty statické a interaktivní vizualizace:

Při rozhodování o výběru přístupu by měla být zohledněna povaha dat, cílová skupina a dostupné zdroje. Statická vizualizace je vhodná tam, kde je potřeba rychlé a jednoduché

Statická vizualizace	Interaktivní vizualizace
Přístup řízený autorem	Přístup řízený čtenářem
Pouze pro čtení	Umožňuje průzkum dat
Jednorázové zobrazení	Vícenásobné vstupy a výstupy
Žádná interaktivita	Vysoká míra interaktivity
Méně flexibilní	Více flexibilní
Lineární přístup	Událostmi řízený styl
Komponenty: data, kód a grafika	Komponenty: data, kód, události a grafika
Neposkytuje nové poznatky	Poskytuje nové poznatky

Tabulka 3.1: Porovnání statické a interaktivní vizualizace [18]

sdělení, zatímco interaktivní vizualizace je ideální pro hlubší analýzy a objevování skrytých souvislostí v datech [7, 18].

3.5 Knihovny pro programátory

Pro vytváření vizualizací dat existuje mnoho různých nástrojů a knihoven, které se liší podle toho, jak s daty a grafikou pracují. Některé umožňují kreslit prvky přímo na stránce, jiné nabízejí pokročilejší možnosti pro práci s daty. K dispozici jsou také knihovny s hotovými šablonami grafů, které vývojářům výrazně usnadňují práci. Tato sekce představí různé přístupy k tvorbě vizualizací a ukáže, kdy se který nástroj nejlépe hodí.

3.5.1 Práce s grafickými prvky na nízké úrovni

Při tvorbě vizualizací na webu je možné začít s technologiemi, které umožňují kreslení jednotlivých tvarů a prvků přímo na stránce. Patří sem především **SVG** a **Canvas**. Tyto nástroje poskytují vývojářům plnou kontrolu nad tím, co se vykreslí, a umožňují vytvářet vizualizace přesně podle vlastních představ. Zároveň ale vyžadují více práce, protože neobsahují žádné hotové funkce pro práci s daty. V této části se podíváme na obě technologie a jejich možnosti.

SVG (Scalable Vector Graphics)

SVG (Scalable Vector Graphics) je výkonný nástroj pro tvorbu interaktivních a dynamic-
kých vizualizací dat. Jedná se o formát založený na XML, který definuje dvourozměrnou grafiku pomocí geometrických tvarů a křivek. Tím se liší od rastrových formátů, jako jsou JPEG nebo PNG, které jsou založeny na jednotlivých pixelech. SVG zajišťuje ostrý a čistý vzhled bez ohledu na velikost nebo rozlišení, což je obzvláště výhodné při prezentaci vizualizací na různých zařízeních [4].

SVG má spoustu výhod, které z něj činí vhodný nástroj pro vizualizaci dat:

- **Nezávislost na rozlišení:** SVG vizualizace zůstávají ostré a čitelné i při zvětšování, což je zásadní výhoda oproti rastrovým formátům jako JPEG nebo PNG [4].
- **Podpora interaktivity:** SVG umožňuje integraci interaktivních prvků, například zobrazení dalších informací při najetí kurzorem na graf [4].

- **Podpora animací:** SVG podporuje animace implementované pomocí CSS nebo JavaScriptu [4].
- **Přístupnost:** Díky atributům jako alt text nebo ARIA labely je možné zpřístupnit vizualizace i uživatelům s vizuálními hendikepy. Tím SVG zajišťuje dostupnost dat širšímu publiku [4].
- **Flexibilita designu:** Díky možnosti oddělit data od jejich vizuální reprezentace můžeme snadno měnit vzhled prvků bez zásahu do samotných dat [3].
- **Jednoduché vytváření složitých struktur:** SVG umožňuje definovat složité datové struktury pomocí XML [3].

Canvas

Canvas je nástroj, který umožňuje vytvářet dynamické a interaktivní grafiky přímo na webových stránkách. Pomocí elementu `<canvas>` z HTML a jazyka JavaScript mohou vývojáři kreslit tvary, text, obrázky nebo vytvářet animace, což poskytuje široké možnosti pro vizualizaci dat [10]. Canvas je oblíbený hlavně proto, že se dá snadno upravit podle potřeb konkrétního projektu [31].

Canvas sdílí některé vlastnosti s SVG, jako je možnost vytvářet interaktivní grafiku a podporovat animace. Obě technologie jsou navrženy tak, aby umožňovaly vizualizaci dat přímo na webových stránkách. Klíčovým rozdílem je ale způsob, jakým obě zpracovávají grafiku. Canvas používá pixely k vykreslení grafiky, což znamená, že výsledné obrazy nejsou nezávislé na rozlišení. Při zvětšení může grafika ztratit kvalitu a stát se rozmazanou. Oproti tomu SVG používá geometrické tvary definované matematickými rovnicemi, což zajišťuje, že grafika zůstane ostrá a čitelná bez ohledu na velikost [11].

Dalším rozdílem je způsob práce s grafikou. Canvas vyžaduje použití JavaScriptu pro vykreslování komplexních obrázků, co z něj dělá vhodnějším pro dynamické aplikace, jako jsou hry nebo vizualizace v reálném čase [11].

Canvas je také efektivnější při práci s velkým množstvím objektů nebo při vytváření složitějších, dynamických a interaktivních grafik. SVG je ideální pro jednodušší grafiku, jako jsou loga nebo tvary, kde je důležitá nezávislost na rozlišení a výkon [11].

3.5.2 Nástroje pro řízení vizualizací nad SVG/Canvas

Jakmile máme k dispozici nástroje jako SVG nebo Canvas, můžeme si práci s daty a kreslením usnadnit pomocí knihoven, které to propojují. Jednou z nich je D3.js, což je knihovna, která umožňuje napojit data přímo na prvky ve vizualizaci. Díky tomu už není potřeba všechno vykreslovat ručně, ale můžeme se soustředit hlavně na to, jak mají data vypadat a jak s nimi uživatel bude pracovat. V této části se podíváme, co D3.js umí a proč se vyplatí ji použít.

D3.js

D3.js (Data-Driven Documents) je populární JavaScriptová knihovna určená pro vizualizaci dat na webu. Byla vytvořena v roce 2011 Mikiem Bostockem a od té doby se stala jedním z nejpopulárnějších nástrojů pro interaktivní vizualizace [6]. D3.js poskytuje programátorům flexibilní nástroje pro vytváření vizualizací s použitím webových technologií, jako jsou HTML, CSS a SVG [29].

Hlavní výhodou D3.js je, že umožňuje pracovat s vizualizací dat od základů. Díky tomu nabízí velké možnosti úprav a úplnou kontrolu nad tím, jak vizualizace vypadá a funguje [16]. D3.js poskytuje různé funkce pro načítání dat, mapování hodnot na vizualizace a tvorbu interaktivních komponent [6].

Mezi hlavní vlastnosti D3.js patří:

- **Flexibilita:** Nabízí velké množství funkcí, které umožňují vývojářům tvořit vlastní vizualizace podle specifických požadavků projektu [6].
- **Podpora webových standardů:** Knihovna využívá HTML, CSS a SVG, což umožňuje snadné používání vizualizací ve webových aplikacích a zajišťuje, že budou fungovat ve všech moderních prohlížečích [29].
- **Interaktivita:** Pomocí D3.js jde vytvářet vizualizace, které reagují na uživatelské akce, například kliknutí nebo najetí myši [16].
- **Modularita:** D3.js je rozdělena do samostatných modulů, které jde používat individuálně nebo kombinovat podle potřeby [6].
- **Animace:** D3.js nabízí vestavěnou podporu pro animace [29].
- **Práce s daty:** Knihovna obsahuje funkce pro práci s datovými sadami, včetně operací nad poli a transformací datových struktur [16].

3.5.3 Vizuální knihovny s hotovými komponentami

Pro případy, kdy není potřeba vytvářet vizualizace úplně od začátku, existují knihovny, které nabízejí předpřipravené komponenty a šablony grafů. Uživatel si zvolí typ grafu, nastaví data a vzhled, a knihovna už se postará o vykreslení. Takové nástroje se velmi hodí při implementaci vizualizací, kde je potřeba rychle zobrazit data v přehledné formě. Mezi známé knihovny tohoto typu patří například Highcharts nebo Chart.js.

Highcharts

Highcharts je knihovna v jazyce JavaScript, která umožňuje snadnou tvorbu grafů pro webové aplikace. Je navržena tak, aby byla jednoduchá na naučení i používání. Uživatelé potřebují jen základní znalosti HTML, CSS a JavaScriptu, aby mohli vytvářet vizualizace podle svých potřeb [23].

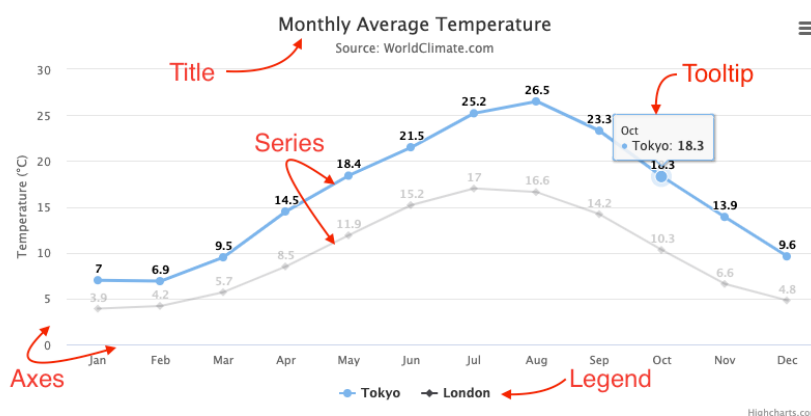
Jednou z výhod Highcharts je rychlé vykreslování grafů a podpora moderních JavaScriptových frameworků, jako jsou Angular, Vue.js nebo React. Grafy můžeme snadno exportovat do různých formátů, jako jsou obrázky, CSV nebo Excel soubory [23].

Highcharts podporuje různé typy grafů, například sloupcové, spojnicové, koláčové, mapové nebo teplotní mapy. Díky velkému množství dostupných funkcí je Highcharts ideální pro tvorbu real-time informačních dashboardů [23].

Struktura grafu v Highcharts zahrnuje několik klíčových částí, jak je znázorněno na obrázku 3.1, který usnadňuje pochopení jednotlivých částí vizualizace:

- **Název (Title):** Popisuje obsah grafu a obvykle se nachází v horní části grafu.
- **Řady (Series):** Představují data zobrazená v grafu, například hodnoty teplot v různých měsících.

- **Nástrojová lišta (Tooltip):** Po najetí kurzorem na datový bod se zobrazí dodatečné informace.
- **Legenda (Legend):** Ukazuje různé datové série a umožňuje je zapínat nebo vypínat.
- **Osy (Axes):** Grafy mají obvykle dvě osy, vodorovnou (x) a svislou (y), které pomáhají měřit a kategorizovat data.



Obrázek 3.1: Základní části grafu v Highcharts [15].

Chart.js

Chart.js je open-source knihovna v JavaScriptu, díky které můžeme snadno vytvářet grafy. Byla vytvořena v roce 2013 a od té doby se stala jednou z nejpoužívanějších knihoven pro vizualizaci dat díky své jednoduchosti a široké podpoře komunitou [8].

Knihovna Chart.js podporuje různé typy grafů, jako jsou sloupcové, spojnicové, radarové a další. Uživatelé si mohou vybrat z mnoha předdefinovaných typů nebo si vytvořit vlastní kombinované grafy [8]. Je snadná na použití, díky tomu i začátečníci mohou rychle vytvořit vizualizace s minimálním kódem [13].

Další výhody Chart.js:

- **Podpora frameworků:** Chart.js je kompatibilní s moderními JavaScriptovými frameworky, jako jsou React, Vue.js nebo Angular [8].
- **Interaktivita:** Chart.js podporuje interaktivní prvky, jako jsou nástroje pro zobrazení podrobností při přejetí myši a klikatelné legendy [13].
- **Responzivní design:** Grafy se automaticky přizpůsobují velikosti obrazovky, což je ideální pro použití na mobilních zařízeních [13].
- **Výkon:** Díky využití technologie HTML5 <canvas> knihovna poskytuje vysoký výkon a rychlé vykreslování i při práci s velkými datovými sadami [8].

Kapitola 4

Analýza problému

Tato kapitola se zaměřuje na analýzu problému spojeného se zpracováním a vizualizací dat z hlasování samospráv. Pozornost je věnována tomu, s jakými daty projekt Zastupko pracuje a jakým způsobem jsou tato data využívána pro vytváření vizualizací. Nejprve se zaměřím na přehled dat, která jsou v systému Zastupko k dispozici, například údaje o zastupitelích, jejich docházce, jednotlivých hlasováních nebo výsledcích zasedání. Dále se podrobně podívám na aktuální vizualizace, které platforma používá. Každý typ vizualizace stručně popíšu a napíšu, jaké znalosti z něj může uživatel získat. Na základě této analýzy se pak zaměřím na to, jaké další znalosti by bylo možné z těchto dat získat, ale které zatím nejsou ve vizualizacích Zastupka. Cílem této analýzy je najít možnosti, jaké nové vizualizace implementovat tak, aby byly pro uživatele ještě přínosnější a lépe vystihovaly chování zastupitelů nebo politických stran, nebo jaké nové vizualizace bude potřeba navrhnout a implementovat.

4.1 Analýza dat

Tato část se zaměřuje na datové zdroje, se kterými Zastupko pracuje. Cílem je přiblížit, jaké informace jsou v datech obsažené, odkud pocházejí a jaké možnosti analýzy a vizualizace nabízejí. Rozdělují je podle toho, zda se jedná o data z městských nebo krajských zastupitelstev.

4.1.1 Data z městských zastupitelstev

Aplikace Zastupko.cz aktuálně využívá otevřená data z měst Brno, Most a Praha. Data jsou dostupná pro dvě volební období (2018–2022 a 2022–2026) a jsou buď přímo z webů měst¹, nebo z portálu otevřených dat². Data obsahují různá informace. Například základní údaje o zastupitelstvu a jeho složení, což znamená seznam všech zastupitelů včetně jména, pohlaví a jejich politické příslušnosti. U každého zastupitele jsou zaznamenány případné změny strany.

Další důležitou částí jsou data o zasedáních zastupitelstva. Ke každému zasedání je uvedeno jeho datum, číslo a seznam projednávaných bodů. U každého bodu je možné dohledat výsledky hlasování, včetně toho, kteří zastupitelé byli přítomní a jak hlasovali (pro, proti, zdržel se, nehlasoval). V datech je také zaznamenáno, jestli bylo hlasování platné a zda bylo schváleno.

¹Například web města Brna: www.bрно.cz

²Portál: <https://data.gov.cz>

Díky těmto informacím je možné z dat získat různé znalosti. Například můžeme spočítat docházku jednotlivých zastupitelů, sledovat, jak často se zdržovali hlasování, nebo analyzovat, jak moc jsou jednotlivé politické strany jednotné při hlasování. Dále je možné porovnávat hlasování mezi zastupiteli nebo sledovat vývoj postojů v čase. Z dat také je možné vyčíst informace o vedení města, například kdo byl primátorem. Celkově tato městská data poskytují poměrně dobrý základ pro různé analýzy i vizualizace a umožňují občanům i analytikům sledovat fungování samospráv přehledněji.

4.1.2 Krajská zastupitelstva

Kromě městských zastupitelstev se zpracovávají i data z krajských zastupitelstev. K dispozici jsou datové sady ze všech čtrnácti krajů včetně hlavního města Prahy. Data o hlasování krajských zastupitelstev pocházejí z oficiálních webů jednotlivých krajů³, nebo z portálu otevřených dat. Některé kraje však poskytují neúplná data nebo data s omezenou strukturou. To znamená, že například některá hlasování nebo zasedání chybí, což může ovlivnit výsledky analýz.

Krajská data jsou docela podobná těm městským. Obsahují informace o zastupitelích, jejich příslušnosti k politickým stranám, přehled zasedání a jednotlivá hlasování. Každé hlasování má informace o tom, jak hlasovali jednotliví zastupitelé, jestli bylo přijato a případně i popis projednávaného bodu. Data umožňují analyzovat docházku zastupitelů, jednotnost hlasování stran nebo poměr přijatých a nepřijatých návrhů.

Zastupko také vizualizuje některé informace v přehledné mapové formě. Například ukazuje, jaký je poměr přijatých a nepřijatých návrhů v jednotlivých krajích nebo kolik hlasování průměrně proběhne během jednoho zasedání. Díky tomu si uživatel může na první pohled všimnout rozdílů mezi kraji. Šedou barvou jsou označeny kraje, kde data chybí nebo nejsou kompletní. Kromě standardních údajů jako je docházka nebo výsledky hlasování, je možné sledovat i takové údaje, jako je počet hlasování na zasedání. Ve srovnání s městy jsou ale krajská data méně konzistentní a některé kraje mají mezery mezi zasedáními. I přesto je možné z těchto dat odvozovat užitečné informace o fungování krajské samosprávy a jejím rozhodování.

4.2 Analýza aktuálních vizualizací v projektu Zastupko.cz

V této části se zaměřím na to, jaké vizualizace jsou aktuálně v Zastupku použity. Popíšu, jaká data jednotlivé vizualizace zobrazují, jakým způsobem jsou informace prezentovány a jaké znalosti si z nich může uživatel vzít. Na základě této analýzy a znalostem z ní odvedených bude později možné navrhnout novou knihovnu vizualizací.

4.2.1 Mandáty

Jednou z vizualizací v projektu Zastupko.cz je znázornění rozdělení mandátů mezi jednotlivé politické strany. Cílem této vizualizace je ukázat, kolik mandátů má která strana nebo koalice v daném zastupitelstvu. Pomocí barevně odlišených koleček je možné snadno zjistit, jaká je síla jednotlivých stran. Na obrázku 4.1 je vidět příklad vizualizace mandátů pro Zastupitelstvo města Brna v období 2022–2026. Každá barva reprezentuje jednu stranu a počet koleček odpovídá počtu mandátů. Uživatel si tak může na první pohled všimnout,

³Například web Jihočeského kraje: www.kraj-jihocesky.cz

která strana má většinu, nebo jak je zastupitelstvo rozloženo. Vizualizace funguje dobře u měst, kde je počet zastupitelů vyšší.

Zastupitelstvo města Brna 2022–2026



Obrázek 4.1: Rozložení mandátů v Zastupitelstvu města Brna (2022–2026)

Na mobilních zařízeních ale může nastat problém u menších rad, například jako je Rada města Mostu (obrázek 4.2). Při nízkém počtu mandátů může rozložení koleček působit nevyváženě nebo být zbytečně roztažené. Bylo vhodné tuto komponentu rozšířit o možnost nastavení počtu řádků a sloupců, aby si vývojář mohl zvolit rozložení vizualizace podle potřeby. Díky tomu by se dala lépe přizpůsobit různým velikostem obrazovek.

Rada města Mostu 2022–2026

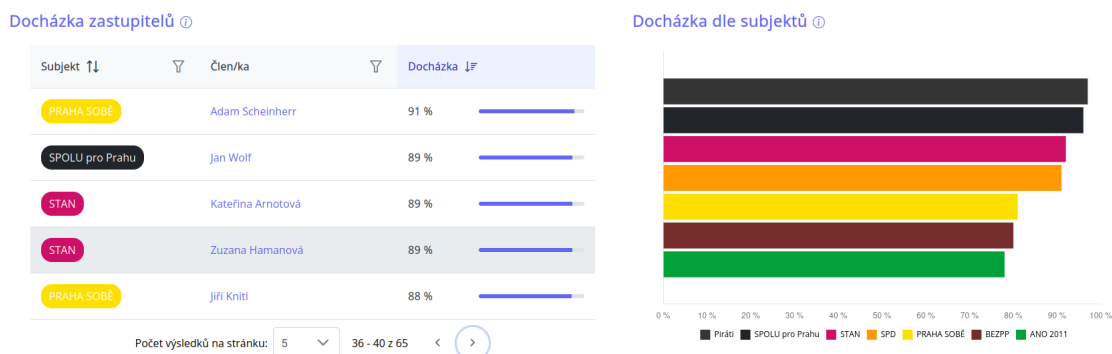


Obrázek 4.2: Rozložení mandátů v Radě města Mostu (2022–2026)

4.2.2 Docházka zastupitelů

Docházka jednotlivých zastupitelů je důležitým důkazem jejich aktivity. Vizualizace na obrázku 4.3 ukazuje docházku zastupitelů v podobě tabulky, kde je u každého jméno, politická strana a procento účasti na zasedáních. Zastupitelé jsou barevně odlišeni podle politických stran. Vedle tabulky je také souhrnný graf, který ukazuje průměrnou docházku podle politických stran. Tímto způsobem jde snadno porovnat, která strana má nejvyšší účast, nebo jestli existují větší rozdíly mezi jednotlivými stranami. Tato vizualizace je přehledná a jde se v ní rychle orientovat, nicméně má i svá omezení. V aktuální podobě není možné sledovat vývoj docházky v čase, například zda se zvyšovala nebo snižovala během volebního

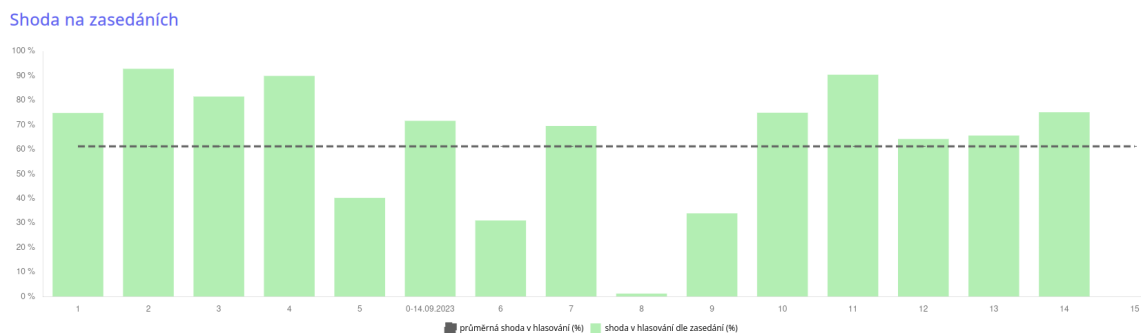
období. Do budoucna by proto bylo vhodné rozšířit analýzu o časovou osu nebo trendovou vizualizaci, která by umožnila lépe sledovat změny chování zastupitelů v čase.



Obrázek 4.3: Vizualizace docházky zastupitelů a průměrné docházky podle stran

4.2.3 Porovnání zastupitelů

Další vizualizace umožňuje porovnávat zastupitele mezi sebou. Uživatel si může zvolit libovolnou dvojici zastupitelů a následně získat přehled o tom, jak často hlasovali stejně nebo odlišně. Výsledkem je vizualizace, která ukazuje míru shody v jejich hlasování na jednotlivých zasedáních, jak je vidět na obrázku 4.4. Sloupce znázorňují procentuální shodu v hlasování mezi vybranými zastupiteli na konkrétním zasedání. Přerušovaná čára pak ukazuje průměrnou shodu mezi všemi zasedáními. Díky tomu jde snadno zjistit, jestli zastupitelé hlasují podobně, nebo jestli se spíš neshodují. Tato vizualizace je velmi užitečná například pro sledování, jak moc zastupitelé shodují s jinými zastupiteli své strany, nebo naopak.

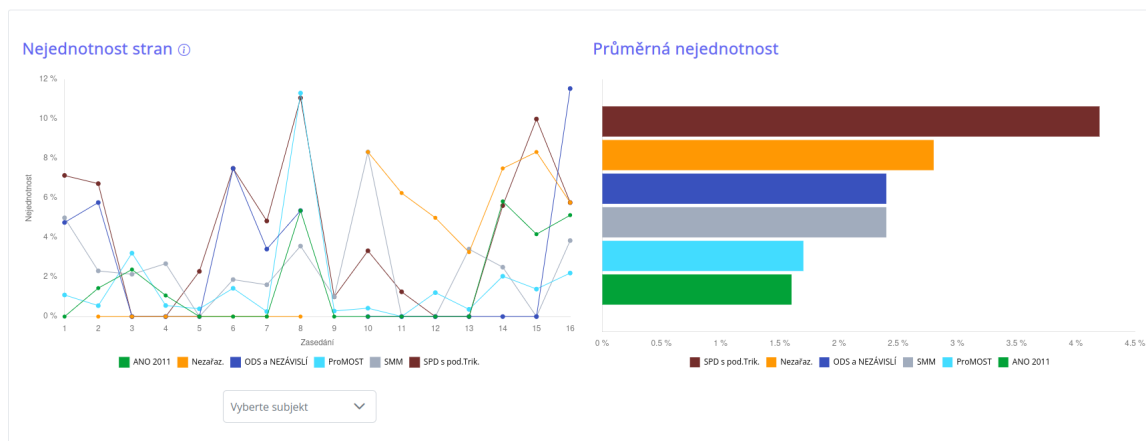


Obrázek 4.4: Shoda v hlasování mezi dvěma vybranými zastupiteli podle jednotlivých zasedání

4.2.4 Nejednotnost stran

Další vizualizace zobrazuje, jak moc jsou jednotlivé strany jednotné při hlasování. To znamená, jestli zastupitelé z jedné strany hlasují většinou stejně, nebo se spíš rozcházejí. Vizualizace ukazuje, jak se nejednotnost měnila na různých zasedáních (vlevo), a také průměrnou nejednotnost za celé období (vpravo), jak jde vidět na obrázku 4.5. Z grafu je vidět, že třeba

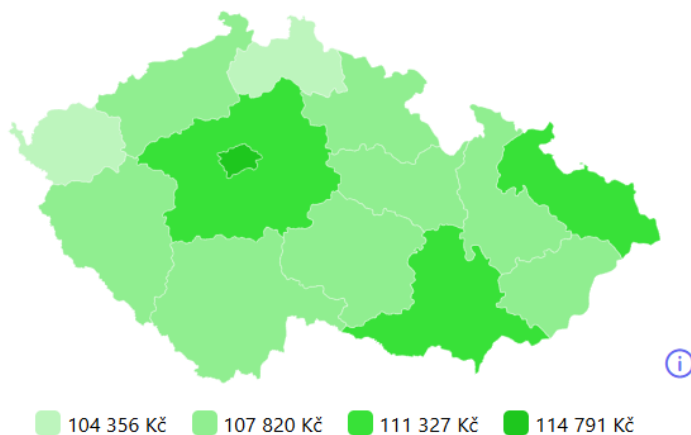
SPD s Trikolorou hlasují nejméně jednotně. Naopak ANO 2011 nebo ProMOST mají mezi svými členy větší shodu. Díky tomu si může uživatel udělat obrázek o tom, jak moc drží strany pohromadě a jestli mají podobné názory. Na rozdíl od vizualizace docházky je tady výhoda, že uživatel může sledovat vývoj v čase a vidět, kdy byla nějaká strana víc rozdělená. To může pomoci odhalit sporná témata nebo období.



Obrázek 4.5: Jak moc jsou strany jednotné při hlasování

4.2.5 Mapové vizualizace

V části s analýzou krajských zastupitelstev jsou také použity mapové vizualizace, jak je vidět na obrázku 4.6. Pomocí mapy si může uživatel snadno porovnat různé hodnoty mezi kraji, například odměny zastupitelů nebo počet hlasování na zasedání. Různé odstíny barev ukazují, jak si který kraj stojí, třeba světlejší barva znamená nižší hodnotu a tmavší vyšší. Po kliknutí na konkrétní kraj se zobrazí sloupcový graf, kde se dají jednotlivé hodnoty přehledně porovnat. Díky tomu si může uživatel udělat lepší představu o tom, jak se kraje liší, aniž by musel procházet složité tabulky. V mapách je možné porovnávat různé typy údajů, nejen odměny. Například jde zobrazit i míru transparentnosti zasedání, počet členů zastupitelstev nebo poměr přijatých a nepřijatých hlasování. To pomáhá rychle zjistit, jak který kraj funguje a kde jsou mezi nimi rozdíly.



Obrázek 4.6: Mapová vizualizace odměn zastupitelů podle krajů

4.3 Nové požadavky na vizualizace

Na základě konzultací s ostatními členy týmu Zastupko vyplynulo, že bude potřeba rozšířit současnou knihovnu o nové typy vizualizací. V diskuzích se ukázalo, že se do projektu plánují přidávat další datové zdroje, například z Českého a Slovenského statistického úřadu nebo z parlamentních hlasování. Tyto nové typy dat přinášejí jiný obsah, a tím pádem i nové požadavky na to, jak je vizualizovat. Zároveň při této analýze vznikl i seznam vylepšení, která by měla zjednodušit práci vývojářům a zlepšit přehlednost výstupů. Diskuze nad požadavky probíhaly postupně během vývoje a byly založeny jednak na zpětné vazbě od ostatních vývojářů, jednak na praktických zkušenostech při vytváření komponent. Na základě této analýzy bylo definováno několik klíčových požadavků:

1. **Možnost předávání dat pomocí funkcí.** Každá komponenta by měla umožňovat předání funkcí, které určují, jak se mají ze vstupních dat získat potřebné informace pro vizualizaci, například jak získat hodnotu, popisek nebo barvu. Tyto funkce se aplikují až uvnitř komponenty, která si data sama zpracuje. Díky tomu není nutné vstupní data předem upravovat a komponenty lze snadno použít i s různě strukturovanými daty.
2. **Konfigurovatelnost přes parametry.** Každá komponenta by měla umožňovat co nejvíce nastavení pomocí parametrů. Vývojář by si měl být schopen upravit například názvy os, pozici nebo zobrazení legendy, ale i to, jakým typem grafu se data vykreslí. Důležité je, aby šlo vizualizaci přizpůsobit konkrétním datům a účelu použití bez potřeby sahát do kódu samotné komponenty. Díky tomu může jedna komponenta sloužit v různých kontextech.
3. **Podpora kombinovaných grafů.** Některé vizualizace by měly umožnit zkombinování více typů grafů, například sloupcového a čárového. Tím je možné zobrazit souvislosti mezi různými metrikami v jednom grafu a zároveň ušetřit místo.
4. **Lepší přizpůsobení pro různé velikosti zařízení.** Vizualizace by měly být responzivní, tedy umět se přizpůsobit jak velikosti obrazovky, tak množství dat. Například u grafu mandátů v Mostu (obrázek 4.2) bylo vidět, že při zobrazení na mobilu s menším počtem koleček rozložení nepůsobilo dobře. Užitečné by bylo umožnit nastavení počtu řádků a sloupců pro zobrazení prvků.

Všechny tyto požadavky vedou k tomu, že nové komponenty nebudou jen jednoduché šablony na vykreslení grafu, ale budou fungovat jako plnohodnotné vizualizační nástroje. Důraz je na jejich přizpůsobitelnost, opakovanou použitelnost a schopnost pracovat s různými daty bez nutnosti měnit jejich vnitřní kód.

Kapitola 5

Návrh nových vizualizačních komponent

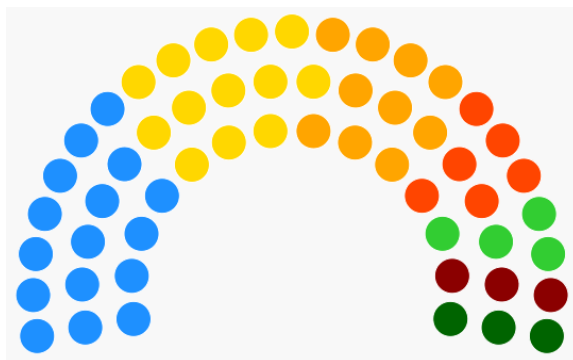
Na základě analýzy dat, které jsem popisoval v předchozí kapitole, je potřeba navrhnout sadu nových vizualizačních komponent. V této kapitole popíšu jednotlivé návrhy těchto vizualizací, které vznikly na základě požadavků od dalších lidí pracujících na projektu Zastupko. Cílem je, aby nové komponenty umožnily lépe vizualizovat nové typy dat a zároveň byly snadno použitelné a přizpůsobitelné podle potřeby. Jednotlivé vizualizace byly zvoleny na základě konkrétních požadavků popsanych v sekci 4.3.

5.1 Vizualizace mandátů

Cílem této vizualizace je přehledně zobrazit rozložení mandátů mezi jednotlivými politickými stranami. V původní verzi projektu Zastupko byly mandáty zobrazeny pomocí jednoduché mřížky koleček, přičemž každé kolečko představovalo jeden mandát. Tento přístup sice umožňoval základní vizualizaci, ale při větším počtu zastupitelů nebyl ideálně čitelný, a proto potřeboval novou verzi.

Nově navržená komponenta proto podporuje dvě varianty zobrazení:

- **Půlkruhová varianta:** Připomíná vizuálně polokruh nebo poloviční koláčový graf. Každá část půlkruhu odpovídá jedné politické straně. Díky tomuto uspořádání si uživatel může snadno představit, jak jsou jednotlivé strany rozloženy v rámci zastupitelstva, podobně jako například v parlamentních diagramech. Tato varianta je přehledná a vhodná pro prezentaci složení zastupitelstva v daném volebním období. Návrh této verze je zobrazen na obrázku 5.1.
- **Mřížková varianta:** Zobrazuje každý mandát jako samostatné kolečko v mřížce. Navazuje na způsob, jakým jsou mandáty v současnosti zobrazovány v systému Zastupko, ale nově bude komponenta výrazně více konfigurovatelná. Vývojář bude moci pomocí parametrů upravit například počet řádků a sloupců nebo pozici legendy. Díky tomu jde mřížku lépe přizpůsobit konkrétním datům i různým velikostem obrazovek.



Obrázek 5.1: Návrh půlkruhového zobrazení mandátů v zastupitelstvu

Vstupem pro vizualizaci je strukturované pole obsahující počty mandátů jednotlivých stran, jejich barvy a zkratky. Výstupem je buď půlkruhový, nebo mřížkový graf v závislosti na nastavených parametrech. Mezi parametry, které lze konfigurovat, patří:

- **isSquare:** Určuje, jestli se má použít mřížková (**true**) nebo půlkruhová (**false**) varianta.
- **partySeats, partyColors, partyAbbreviations:** Pole s daty o jednotlivých stranách.
- **rows, cols:** Počet řádků a sloupců pro mřížkovou variantu.
- **legendBelow:** Pozice legendy (pod grafem nebo vedle).

Důraz je kladen na responsivitu. Legenda se může podle nastavení nebo velikosti obrázky přesunout pod graf a velikost jednotlivých prvků se mění v závislosti na jejich počtu. Komponenta by tak měla být použitelná jak na desktopu, tak na mobilních zařízeních.

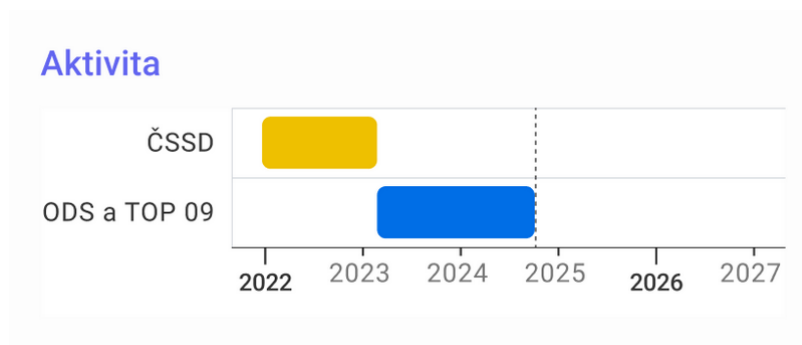
5.2 Časová osa aktivit

Další požadovanou vizualizací je časová osa aktivit. Cílem této vizualizace je přehledně zobrazit, v jakých letech byli zastupitelé aktivní v jednotlivých stranách. Na obrázku 5.2 je příklad, jak by tato vizualizace mohla vypadat. Každý řádek odpovídá jedné straně a barevné pruhy ukazují období, kdy byli členové této strany aktivní. Díky tomu si uživatel může snadno udělat představu o tom, kdy daný zastupitel působil v jednotlivých stranách a jak dlouho to trvalo.

Vstupem do komponenty je pole objektů, z nichž každý obsahuje název strany, začátek a konec období v letech a barvu, kterou má být daný úsek vykreslen. Vizualizace zároveň podporuje parametry jako orientace (horizontální nebo vertikální), zobrazení čáry pro aktuální datum a automatické přizpůsobení velikosti.

V budoucnu ale nemusí sloužit jen pro aktivity stran. Stejnou vizualizaci by se dalo využít i pro jiné časové údaje. Proto je důležité navrhnout komponentu dostatečně univerzálně, aby šla přizpůsobit různým datovým vstupům. Jedním z požadavků je, aby časová osa obsahovala čáru pro aktuální datum. Ta by měla uživatelům pomoci lépe se zorientovat v tom, co už je minulost a co je současnost. Díky této čáře bude vizualizace přehlednější. Další klíčovou vlastností je možnost přepínat mezi vertikálním a horizontálním zobrazením. V některých případech bude vhodnější klasické horizontální rozložení, kdy čas běží

zleva doprava. Jindy, například na mobilních zařízeních, bude lepší vertikální uspořádání. Vizualizace by měla být schopná přizpůsobit se podle nastavení parametrů.

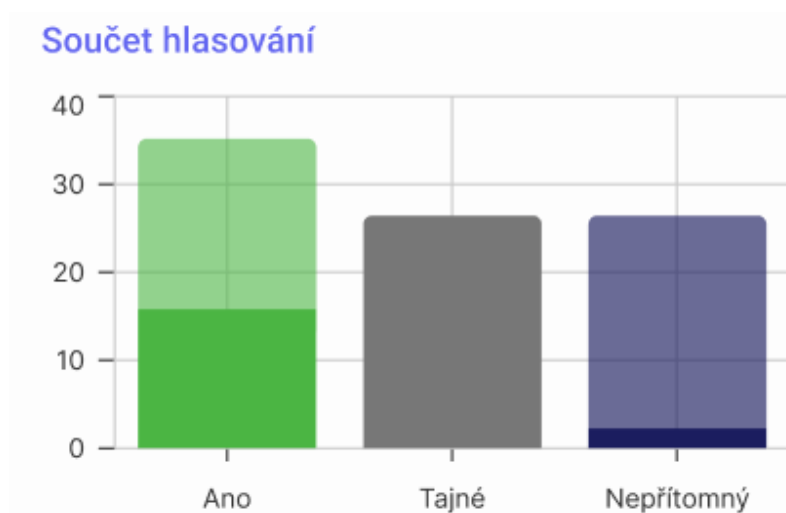


Obrázek 5.2: Návrh vzhledu časové osy aktivit

5.3 Sloupcový graf

Další požadovanou vizualizací je sloupcový graf. Při diskuzích s Zastupk týmem vyplynulo, že bude potřeba zobrazovat různé typy sloupcových grafů v různých situacích. Aby nebylo nutné vytvářet pokaždé novou komponentu, bylo potřeba navrhnout univerzální vizualizaci, která bude co nejvíce přizpůsobitelná pomocí parametrů. Pomocí nastavení bude možné například přepínat mezi vertikálním a horizontálním zobrazením, což umožní přizpůsobit graf typu zařízení. Další funkcí bude zvýraznění sloupců, které splňují určitou podmínku, například překračují danou hodnotu, nebo mají maximální hodnotu. Speciálním režimem je varianta s průhledností, kde budou sloupce vykresleny s různou intenzitou barvy podle výšky hodnoty. Ukázka tohoto režimu je vidět na obrázku 5.3. Vizualizace je navržena jako univerzální komponenta s několika konfiguračními parametry, které určují její chování a vzhled:

- **data:** Pole objektů, které obsahují **label**, **value**, případně **total** a **color**.
- **isVertical:** Určuje orientaci grafu (vertikální nebo horizontální).
- **highlightAbove:** Volitelná hodnota, nad kterou se sloupce zvýrazní.
- **highlightMax:** Parametr, který určuje, jestli zvýraznit sloupec s nejvyšší hodnotou.
- **opacityVersion:** Přepíná režim s průhledností sloupců (viz obrázek 5.3).



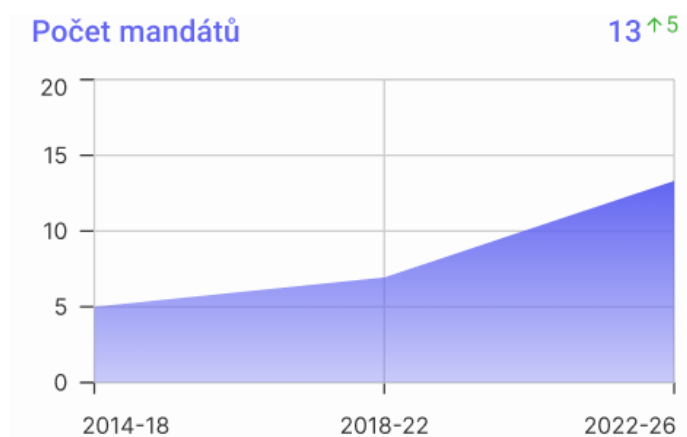
Obrázek 5.3: Ukázka návrhu režimu s průhledností sloupcového grafu

5.4 Čárový graf

Podobně jako u sloupcového grafu se ukázalo, že i u čárových grafů existuje více různých požadavků podle typu zobrazených dat. Proto bylo potřeba navrhnout univerzální vizualizaci, která půjde pomocí parametrů snadno přizpůsobit konkrétnímu použití. Jedním z hlavních požadavků je možnost změnit orientaci grafu, aby bylo možné jej vykreslit vertikálně například na mobilních zařízeních. Dále by graf měl podporovat zobrazení s vyplněnou plochou pod čarou, což pomůže zvýraznit vývoj hodnot v čase. Pokud je zapnutý parametr pro vyplnění, barva pod čarou je tvořena gradientem, jak ukazuje obrázek 5.4. Při zobrazení více čar lze tento režim jednoduše vypnout, aby nedocházelo k překryvu. Dále by mělo být možné každé čáře nastavit vlastní barvu a tloušťku čáry.

Tato vizualizace by měla sloužit jako univerzální komponenta pro zobrazení časových dat v různých kontextech, například při porovnání vývoje mezi stranami nebo sledování účasti na zasedáních. Umožňuje konfiguraci následujících parametrů:

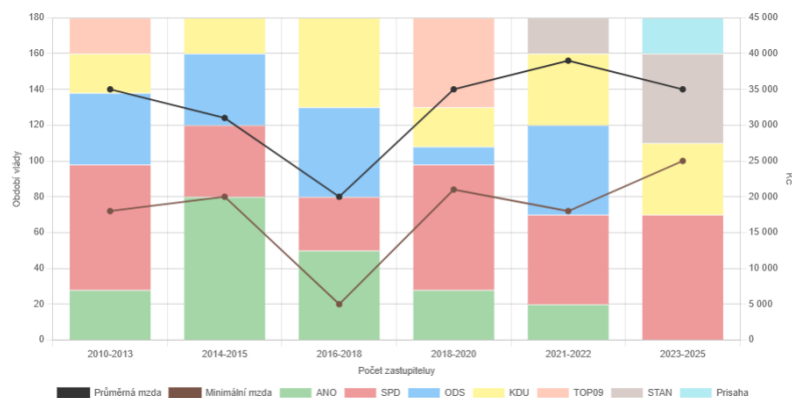
- **data:** Vstupní datová struktura buď jako pole sérií s hodnotami, nebo jako objekt s funkcemi.
- **isVertical:** Nastavuje orientaci os (standardně horizontální, po zapnutí vertikální).
- **isFilled:** Pokud je aktivní, vyplní oblast pod čarou barevným gradientem (viz obrázek 5.4).



Obrázek 5.4: Ukázka čárového grafu s barevným vyplněním pod čarou

5.5 Kombinovaný graf

Protože se do projektu Zastupko plánují přidávat další datové zdroje, roste i množství různých informací, které je potřeba přehledně zobrazit. Často nestačí vizualizovat pouze jeden typ dat, ale je vhodné vidět i jejich vzájemné souvislosti. Z toho důvodu vznikla potřeba vytvořit komponentu, která spojí dva typy grafů, sloupcový a čárový, do jednoho. Výsledný kombinovaný graf umožňuje například znázornit počty zastupitelů pomocí složených sloupců a zároveň vykreslit vývoj určité metriky, jako je průměrná mzda znázorněná čarou. Ukázka takového zobrazení je uvedena na obrázku 5.5.



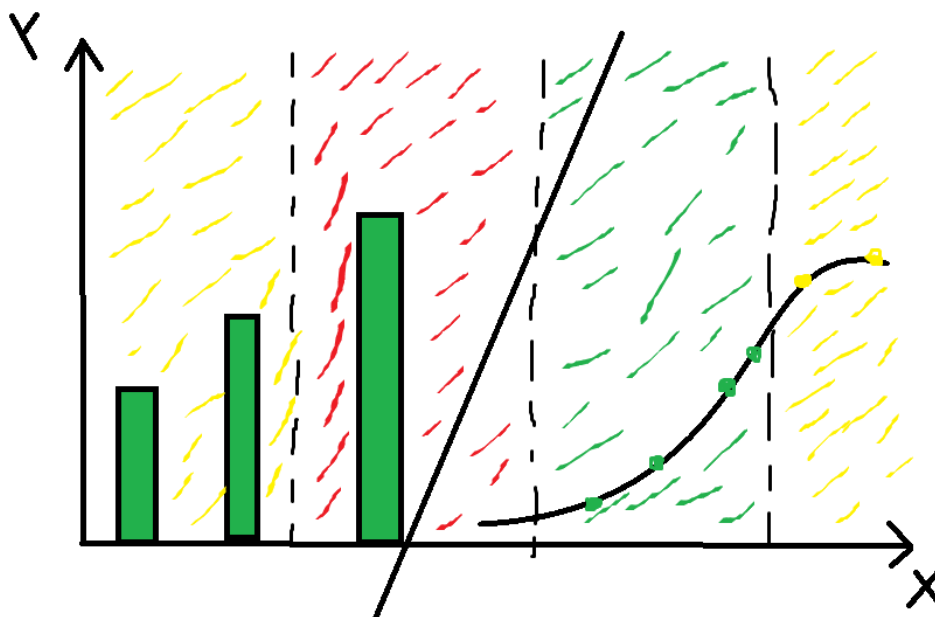
Obrázek 5.5: Ukázka kombinovaného grafu

Aby byla komponenta univerzální, jednotlivé části grafu (sloupce i čáry) jsou generovány pomocí funkcí, které vývojář předá jako parametry. Lze tak jednoduše určit, které části dat se mají vykreslit jako skládané sloupce a které jako čáry. Komponenta podporuje i přepínání orientace grafu mezi vertikální a horizontální. Vizualizace očekává vstup ve formátu pole objektů, kde každý objekt reprezentuje jednu kategorii. Každý záznam obsahuje jak informace pro sloupce, tak i data pro čárovou část grafu. Vstupní data mají následující strukturu:

- **label:** Textový popis osy x.
- **name:** Seznam názvů kategorií (např. názvy politických stran).
- **values:** Pole číselných hodnot odpovídajících položkám v poli **name**.
- **color:** Pole barev odpovídajících jednotlivým kategoriím ve složených sloupcích.
- **lineCharts:** Pole objektů s hodnotami pro čárový graf; každý objekt obsahuje název (**name**), hodnotu (**value**) a barvu (**color**).
- **isHighlighted:** Nepovinný parametr určující, jestli má být daný sloupec zvýrazněn.

5.6 Vizualizace s podporou období

Na základě požadavků uvedených v kapitole 4 byla navržena vizualizace s možností zvýraznění časových období přímo na pozadí grafu. Cílem je, aby si uživatelé mohli lépe všimnout, kdy se v datech odehrávaly nějaké důležité změny nebo události. Ukázku toho, jak by to mohlo vypadat, je možné vidět na obrázku 5.6.



Obrázek 5.6: Ukázka grafu s barevně vyznačenými obdobími

Vývojář by měl mít možnost si nastavit, kde každé období začíná a končí, a také jakou barvou se dané období zobrazí. Data by měla být zobrazena jak pomocí sloupcového grafu, tak pomocí čárového grafu, nastaví se to pomocí parametru. Na rozdíl od některých jiných grafů bude tato vizualizace podporovat pouze horizontální variantu. To znamená, že časová osa bude vždy na ose X, protože právě k ní se vážou jednotlivé periody. Díky tomuto rozšíření bude možné uživatelům lépe přiblížit, co se v datech dělo v různých obdobích, aniž by museli složitě dohledávat jednotlivé časové úseky. Vstupní parametry komponenty jsou následující:

- **data:** Pole datasetů obsahující popisek, hodnoty a barvu.
- **years:** Seznam všech roků, které tvoří osu x.
- **periods:** Pole objektů, z nichž každý obsahuje:
 - **start:** Začátek období.
 - **startOffset:** Jemné posunutí začátku (0–1).
 - **end:** Konec období.
 - **endOffset:** Jemné posunutí konce (0–1).
 - **color:** Barva období.
- **chartType:** Typ grafu, buď „**bar**“ nebo „**line**“.
- **separatePeriods:** Jestli se mají kreslit ohraničení období pomocí čárování.
- **fontSize:** Velikost písma pro popisky období.

5.7 Mapová vizualizace

Poslední navrhovanou komponentou je obecná mapová vizualizace. V současné době existuje v projektu Zastupko mapa České republiky rozdělená na jednotlivé kraje, kde je každý kraj samostatné SVG. Cílem nové komponenty ale bude vytvořit obecnější řešení. Komponenta by měla pracovat s daty ve formátu GeoJSON. Pomocí předaného GeoJSON souboru by měla umět zobrazit jakoukoli mapu, například mapu celé České republiky, Slovenska, nebo obou států dohromady. Vizualizace by měla být univerzální, aby se dala v budoucnu snadno použít i pro jiné mapy.

Jednou z hlavních funkcí této komponenty bude možnost vybrat, které části mapy se mají zobrazit. Pokud bude GeoJSON obsahovat například data pro Českou republiku i Slovensko, vývojář bude moci pomocí parametru určit, jestli chce zobrazit jen Českou republiku, jen Slovensko, nebo obě země zároveň. To umožní lepší práci s daty a větší flexibilitu při vytváření různých typů vizualizací. Výsledná komponenta tak umožní jednoduše zobrazit různé geografické oblasti podle potřeby a navíc bude připravena na případné rozšíření o další regiony nebo nové typy dat.

Kapitola 6

Implementace

V této kapitole popíšu, jak jsem jednotlivé vizualizace z návrhu skutečně vytvořil. Budu se věnovat každé komponentě zvlášť a vysvětlím, jak jsem ji implementoval.

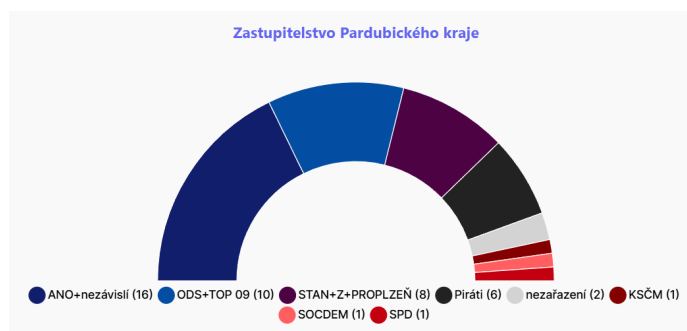
6.1 Vizualizace mandátů

Komponenta `ParliamentChartWidget` implementuje dvě verze grafu, půlkruhovou a mřížkovou. Původně byla zvažována knihovna `Highcharts` pro vykreslení půlkruhu, ale kvůli velikosti knihovny byla nakonec použita knihovna `Chart.js`, která tuto vizualizaci umožňuje pomocí upraveného koláčového grafu. Půlkruhový graf je realizován pomocí typu `doughnut` s upravenými parametry `rotation`, `circumference` a `cutout`. Příklad konkrétní konfigurace je uveden ve výpisu 6.1.

```
type: 'doughnut',
options: {
  rotation: -90,
  circumference: 180,
  cutout: '60%',
}
```

Výpis 6.1: Konfigurace půlkruhového grafu v `Chart.js`

Pro zvýrazňování částí grafu při najetí myši se využívá vlastní logika s pomocí funkce `setActiveElements`, která dočasně zvýrazní odpovídající úsek grafu. Součástí vykreslení je i vlastní generátor legendy, který nahrazuje výchozí styl `Chart.js`, aby byl graf lépe sladěn se zbytkem aplikace. Výsledná podoba grafu je zobrazena na obrázku 6.1.



Obrázek 6.1: Půlkruhová vizualizace mandátů v `Chart.js`

V případě, že je aktivována mřížková varianta, komponenta se vykresluje čistě pomocí HTML a CSS bez použití externí knihovny. Pro rozložení prvků se používá CSS grid. Nejprve se na základě vstupních dat vytvoří pole všech mandátů. Poté se pro každý mandát vytvoří element `div` jazyka HTML, kterému se přiřadí odpovídající barva podle stranické příslušnosti.

Interaktivita funguje pomocí událostí `onmouseenter` a `onmouseleave`. Při najetí na jedno z koleček se zvýrazní všechny mandáty dané strany. Součástí komponenty je také vlastní legenda, která se může zobrazit vedle vizualizace nebo pod ní, v závislosti na nastavení a velikosti obrazovky. Při menší šířce se legenda automaticky přesune pod graf, aby byl výstup čitelnější na mobilních zařízeních. Velikost jednotlivých koleček se počítá dynamicky podle počtu prvků.

6.2 Časová osa aktivit

Pro komponentu časové osy aktivit jsem nenašel žádnou knihovnu, která by splňovala požadavky, proto jsem se rozhodl vytvořit ji pomocí knihovny D3.js. V komponentě se vytvoří SVG prvek, do kterého se pomocí D3 vykreslí barevné obdélníky podle zadaných období. Pokud je parametr `isVertical` nastaven na `true`, v tomto režimu osa Y reprezentuje roky a osa X jednotlivé strany. Pro určení přesných pozic se použijí dvě funkce: `scaleLinear` pro roky (aby bylo možné převést čas na výšku obdélníku) a `scaleBand` pro strany (aby se rozložily rovnoměrně do šířky grafu). Pro každé období se potom vykreslí obdélník, který znázorňuje dané roky. Barva se vezme z vstupních dat a velikost i pozice bloku se počítají podle těchto škál. Vykreslení jednotlivých období pomocí SVG obdélníků je realizováno kódem uvedeným ve výpisu 6.2.

```
svg.selectAll(".bar")
  .data(data)
  .enter()
  .append("rect")
  .attr("x", d => xScale(d.party))
  .attr("y", d => yScale(d.endYear))
  .attr("height", d => yScale(d.startYear) - yScale(d.endYear))
  .attr("width", xScale.bandwidth())
  .attr("fill", d => d.color)
```

Výpis 6.2: Vykreslení období jako barevných obdélníků v D3.js

Tento blok vezme všechna data a pro každou položku vytvoří SVG obdélník. Vodorovná pozice obdélníku (`x`) závisí na tom, za jakou stranu zastupitel působil. Svislá pozice a výška (`y` a `height`) jsou odvozeny z let začátku a konce členství, takže obdélník se natáhne přes celou délku období. Barva se určí podle vstupního parametru `color`, aby bylo možné rozlišit různé strany. Když uživatel najede myší na obdélník, komponenta zobrazí informace s podrobnostmi, jako jsou název strany a konkrétní roky. Dále komponenta pomocí tzv. hooků sleduje velikost kontejneru a při každé změně velikosti automaticky překreslí graf.

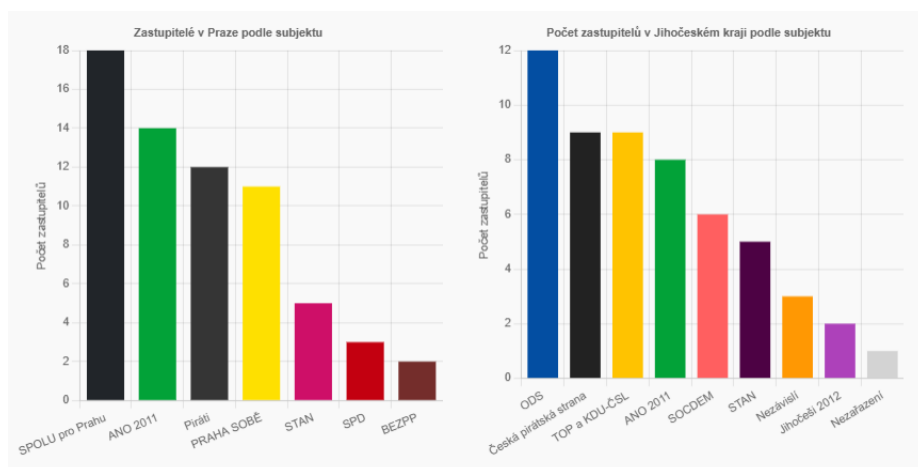
6.3 Sloupcový graf

Pro implementaci této vizualizace jsem použil knihovnu Chart.js. Komponenta se jmenuje `BarChartWidget` a podporuje několik konfiguračních režimů, které se přepínají pomocí parametrů. Data se nejprve připravují z pole objektů, případně přes funkci `getValues()`.

V případě aktivního parametru `opacityVersion` se vytvoří dvě datové sady: první představuje reálnou hodnotu a druhá zbytek do celkové hodnoty s menší průhledností. Obě sady se pak vykreslí jako stohované sloupce. Pokud je průhlednost vypnutá, zobrazí se pouze jedna sada hodnot.

Zvýraznění sloupců se řeší pomocí `borderColor` a `borderWidth`, kde se dynamicky nastavuje obrys u sloupců, které splňují podmínku pro zvýraznění. Barva rámečku se nastavuje na černou nebo červenou podle typu zvýraznění.

Orientace grafu se nastavuje pomocí vlastnosti `indexAxis`, která je přepínána podle parametru `isVertical`. Graf se vždy přizpůsobuje dostupné velikosti kontejneru a má vypnutou legendu. Finální podobu komponenty ukazuje obrázek 6.2.



Obrázek 6.2: Výsledné zobrazení sloupcového grafu

Výchozí orientace grafu je vertikální. Pokud je nastaven parametr `isVertical` na `false`, osa se přepne a graf se zobrazí horizontálně. To je vhodné například v případě, kdy jsou popisky dlouhé nebo je třeba porovnat více kategorií. Komponenta je navržena tak, aby byla plně responzivní a vždy se přizpůsobila velikosti kontejneru.

6.4 Čárový graf

Podobně jako sloupcový graf, i čárový graf využívá knihovnu Chart.js. Komponenta se jmenuje `LineChartWidget` a její implementace je navržena tak, aby šla snadno znovu použít na různých místech. Na vstupu přijímá buď pole dat, nebo objekt s metodami jako `getValues`, `getParties` nebo `getColor`. Nejdříve komponenta z dat vytvoří popisy a pak si připraví dataset pro každou čáru, která se má vykreslit. Každá čára má svoji barvu, popis a pole hodnot. Barva se použije jako `borderColor`. Pokud je aktivní režim `isFilled`, tak se použije i jako `backgroundColor`. Komponenta poskytuje možnost vyplnění pod čarou. Pokud je parametr `isFilled` zapnutý, tak se vykreslí barevný přechod od barvy čáry až po průhlednou. Barevný přechod pro výplň pod čarou je generován pomocí funkce `getGradient`, jejíž ukázka je uvedena ve výpisu 6.3.

```
const gradient = ctx.createLinearGradient(0, area.top, 0, area.bottom);
gradient.addColorStop(0, baseColor);
gradient.addColorStop(1, `${baseColor}00`);
```

Výpis 6.3: Vytvoření barevného přechodu pro výplň pod čarou

Tahle funkce se použije v `backgroundColor` přes funkci, která dostane kontext grafu. Výplň je nastavená tak, že se vykresluje směrem dolů k ose. Pro plynulejší vzhled je nastavena i vlastnost `tension`, která trochu zaoblí rohy mezi body, a také `spanGaps`, takže graf spojí body i v případě, že některé hodnoty chybí. Na obrázcích 6.3a a 6.3b je vidět, jak se čárový graf chová při různém nastavení parametrů, například při zapnutém vyplnění oblasti pod čarou, nebo při zobrazení více sérií v vertikální orientaci.



(a) Vertikální čárový graf s více čarami



(b) Čárový graf s výplní pod čarou

Obrázek 6.3: Ukázka různých variant čárového grafu podle nastavení parametrů

6.5 Kombinovaný graf

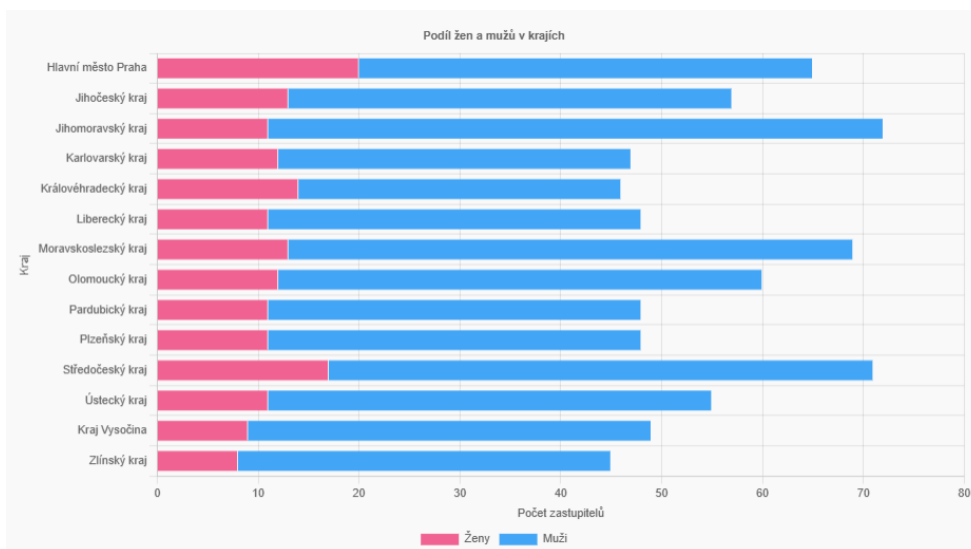
Komponenta `StackedBarWithLineChart` kombinuje složený sloupcový graf a volitelně i čárový graf do jednoho grafu. Je napsaná pomocí knihovny `Chart.js`. Hlavní logika se skládá ze dvou částí. Nejdřív se vytvoří datové řady pro složené sloupce. Každý sloupec má svůj vlastní dataset, který se vloží do jednoho společného zásobníku. Data se vytvářejí dynamicky, to znamená, že pro každou kategorii komponenta projde všechny řádky a zjistí, jestli se daná strana v daném řádku vyskytuje, a pokud ano, použije odpovídající hodnotu. Ukázka způsobu generování datových řad pro složené sloupce je uvedena ve výpisu 6.4.

```
bars.push({
  label: name,
  data: rows.map((r, i) => {
    const idx = getNames(rows, r, i).indexOf(name);
    return idx !== -1 ? getVals(rows, r, i)[idx] : 0;
  }),
  ...
});
```

Výpis 6.4: Vytvoření datové řady pro složené sloupce

Ve druhém kroku se přidají čárové hodnoty, pokud jsou ve vstupních datech. Čáry jsou vykresleny přes samostatnou osu `y1`, která se přidá jen v případě, že je graf ve vertikálním režimu a nějaké čárové hodnoty existují. Každá čára má své jméno, barvu a hodnoty pro

jednotlivé záznamy. Pokud existuje více čar, komponenta je oddělí podle indexu a zajistí, že budou ve správném pořadí. Výsledná data se pak spojí do jednoho pole `datasets`, kde jsou nejdřív čáry (typ `line`) a pak sloupce (typ `bar`). Všechny tyto části mají nastavené pořadí vykreslení přes atribut `order`, aby čára byla vždy navrchu. Při implementaci této komponenty jsem se inspiroval přímo v oficiální dokumentaci knihovny Chart.js¹. Příklad použití komponenty v horizontální variantě komponenty je vidět na obrázku 6.4.



Obrázek 6.4: Ukázka horizontální varianty kombinovaného grafu

6.6 Vizualizace s podporou období

Komponenta `VotingPeriodChart` rozšiřuje klasický čárový nebo sloupkový graf o možnost zobrazit v pozadí jednotlivá období, která se vykreslují jako barevné, poloprůhledné pruhy přes osu x. Periody se předávají jako pole nebo jako objekt s metodami, které vrací začátek a konec období, případně barvu a posuny (`startOffset`, `endOffset`). Nejdůležitější část vykreslování období je plugin Chart.js `beforeDraw`. V něm se nejprve určí, odkud a kam se má každé období kreslit. Výpočet pozice každého období na ose X je ilustrován ve výpisu 6.5.

```
let xStart = x.getPixelForValue(allLabels[startIndex]);
let xEnd = x.getPixelForValue(allLabels[endIndex]);

const stepSize = x.getPixelForValue(allLabels[startIndex + 1]) -
                 x.getPixelForValue(allLabels[startIndex]);

xStart += period.startOffset * stepSize;
xEnd += period.endOffset * stepSize;
```

Výpis 6.5: Výpočet pozice období v pixelech

Tohle určuje počáteční a koncovou pozici období v pixelech, podle pozice na ose X. Offsety umožňují jemné doladění, např. aby období nezačínalo přesně na začátku roku, ale třeba v jeho polovině. Situace, kdy by období přesahovalo zobrazený rozsah grafu, je řešena podmínkami uvedenými ve výpisu 6.6.

¹<https://www.chartjs.org/docs/latest/samples/other-charts/combo-bar-line.html>

```
if (period.start <= minLabel && period.startOffset === 0) xStart = x.left;  
if (period.end >= maxLabel && period.endOffset === 1) xEnd = x.right;
```

Výpis 6.6: Omezení vykreslení období na rozsah osy

Díky tomu se nestane, že by se období vykreslilo mimo graf nebo jen částečně. Vlastní vykreslení období do pozadí grafu je realizováno pomocí obdélníku, jak ukazuje výpis 6.7.

```
ctx.fillStyle = fillColor;  
ctx.globalAlpha = 0.3;  
ctx.fillRect(xStart, y.top, xEnd - xStart, y.bottom - y.top);  
ctx.globalAlpha = 1;
```

Výpis 6.7: Vykreslení pozadí jako barevného pruhu

Barva je nastavená na tu, která je u daného období zadaná, a průhlednost na 30 %. Plugin se přidává do grafu jako součást objektu `plugins` a funguje pro oba typy grafů. Komponenta tak umožňuje vizualizaci vývoje dat v čase spolu s přehledným vyznačením důležitých období. Logika registrace pluginu a vykreslování do pozadí grafu je inspirována dokumentací Chart.js².

6.7 Mapová vizualizace

Mapová vizualizace je řešena pomocí komponenty `RegionalChoroplethWrapper`, která slouží jako jednoduchá obálka (tzv. wrapper) nad již existující komponentou `RegionalChoropleth`. Jejím hlavním účelem je připravit geografická data tak, aby je bylo možné přímo předat do mapové komponenty. Pokud je aktivní parametr `geojsonFormat`, komponenta provede převod dat z GeoJSON formátu do podoby, kterou očekává `RegionalChoropleth`. Tento převod probíhá pomocí knihovny D3.js, která poskytuje mnoho funkcí pro práci s takovými daty³, konkrétně přes projekci `geoMercator` a generátor cest `geoPath`, které vygenerují SVG cestu (`d`) pro každou oblast. Hlavní část převodu vypadá takto:

```
const projection = d3.geoMercator().fitSize([width, height], filteredGeoJson);  
const pathGenerator = d3.geoPath().projection(projection);  
  
geo[regionId] = {  
  name,  
  polygon: { d: pathGenerator(feature) },  
  capital: undefined  
};
```

Výpis 6.8: Převod GeoJSON dat na SVG cestu pomocí D3.js

Pomocí této části kódu se každá oblast převede na vykreslitelný SVG tvar. Kromě toho komponenta umožňuje filtrování oblastí přes parametr `regionFilter`, který omezí, které oblasti se mají v mapě zobrazit. Výsledná data se následně předají komponentě `RegionalChoropleth`, která se postará o samotné vykreslení mapy, vybarvení oblastí podle hodnot a zobrazení legendy. Komponenta sama o sobě žádné vykreslování neřeší, pouze připraví data ve správném tvaru.

²Dokumentace Chart.js: <https://www.chartjs.org/docs/latest/getting-started/usage.html>

³Dokumentace D3.js: <https://d3js.org/d3-geo>

Kapitola 7

Testování

Testování vizualizačních komponent probíhalo ve třech navazujících fázích. Nejprve proběhlo testování na lokálním vývojovém prostředí, kde byly ověřovány základní funkce a chování komponent při různých vstupech. Poté následovalo ověření komponent přímo v systému Zastupko.cz, kde bylo možné pracovat s reálnými daty krajských zastupitelstev. Nakonec byly komponenty využity v dalších bakalářských pracích, kde se ukázalo jejich praktické použití.

7.1 Lokální testování

V první fázi jsem všechny komponenty testoval lokálně na svém počítači. Používal jsem lokální server a testovací sadu dat ve formátu JSON. Cílem bylo ověřit, že vizualizace správně fungují s různými vstupy a konfiguracemi.

Každou komponentu jsem testoval s několika sadami dat. Vyzkoušel jsem například situace, kdy chybí některé očekávané hodnoty nebo kdy pole obsahují prázdné položky. Dále jsem testoval přepínání parametrů, které mění vzhled a strukturu vizualizace, například změnu popisků os nebo přítomnost nebo nepřítomnost legendy. Pro jednodušší generování různých variant testovacích dat jsem využil nástroj ChatGPT¹.

Důraz jsem dával i na odhalení chyb při předávání dat pomocí funkcí, což je klíčový požadavek. Při tomto testování jsem narazil na řadu drobných chyb, které jsem průběžně opravoval, například špatně zpracované prázdné hodnoty, špatné zarovnání legendy při menší šířce obrazovky nebo chyby při kombinaci více grafových prvků.

7.2 Testování v projektu Zastupko.cz

Po úspěšném ověření základních funkcí jsem testoval přímo v systému Zastupko.cz. Vytvořil jsem si vlastní testovací dashboard, kde jsem mohl komponenty propojit s reálnými daty z krajských zastupitelstev. Testoval jsem data ze všech krajů, která obsahovala data jako počet obyvatel, průměrnou hrubou mzdu, nezaměstnanost nebo vývoj kriminality v letech 2020 až 2023. Každý kraj měl desítky hodnot v různých časových obdobích, což umožnilo ověřit fungování vizualizací i pro rozsáhlejší datové sady.

Použil jsem komponenty pro čárové, sloupcové a kombinované grafy a sledoval, zda se data zobrazují správně a přehledně. Díky tomu jsem měl jistotu, že jsou připravené na reálné nasazení a mohou být využity i dalšími vývojáři v rámci systému.

¹<https://chat.openai.com>

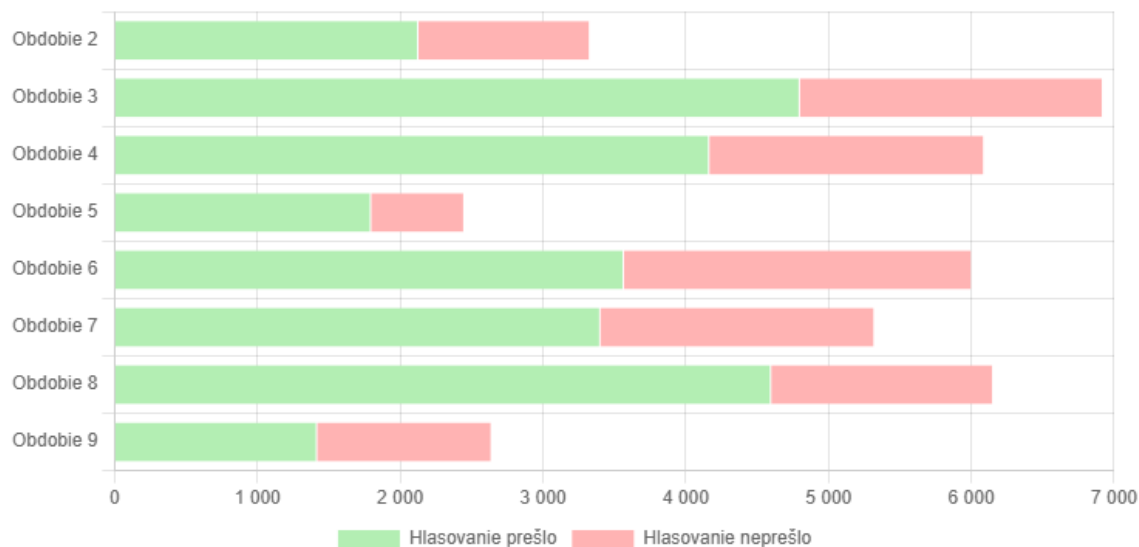
7.3 Praktické použití vizualizací

Poslední fáze testování proběhla formou praktického nasazení komponent v dalších bakalářských pracích, které vznikaly v rámci systému Zastupko.cz. Konkrétně šlo o dvě bakalářské práce. První z nich analyzovala data z Českého statistického úřadu a slovenského Štatistického úřadu [27], druhá se zaměřila na analýzu dat z hlasování Parlamentu České republiky [17].

Obě tyto práce potřebovaly vhodný způsob vizualizace výsledků svých analýz. Využily k tomu komponenty vytvořené v rámci této práce, které umožnily zobrazit výstupy ve srozumitelné grafické podobě. V této fázi se ukázalo, že komponenty jsou skutečně univerzální a je možné je použít pro zobrazení různých informací. Stačilo je pouze nakonfigurovat podle daného typu dat.

První práce [27] využila komponenty vytvořené v této práci k vizualizaci různých datových výstupů. Příklady těchto vizualizací jsou uvedeny na obrázcích 7.1 a 7.2. Druhá práce [17], která se zaměřila na parlamentní hlasování, rovněž využila všechny vizualizační komponenty vytvořené v rámci této práce. Mezi jinými zde byl použit i kombinovaný graf, například pro zobrazení vývoje metrik při různém složení vlád. Na obrázku 7.3 je uveden konkrétní příklad čárového grafu zobrazujícího vývoj jedné z metrik v čase.

Počet hlasování podľa volebních období



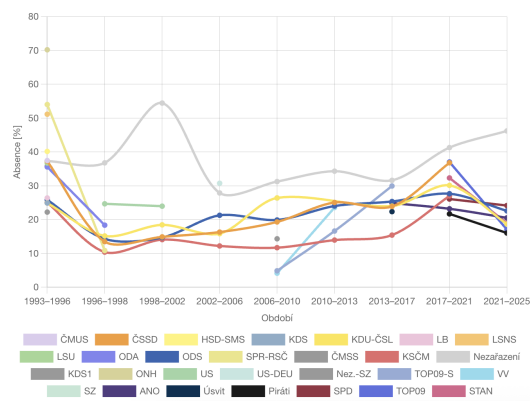
Obrázek 7.1: StackedBar vizualizace hlasování podle volebních období [27]



Obrázek 7.2: Čárový graf vývoje souhlasnosti jednotlivých poslaneckých stran [27]

Absence podle stran (%)

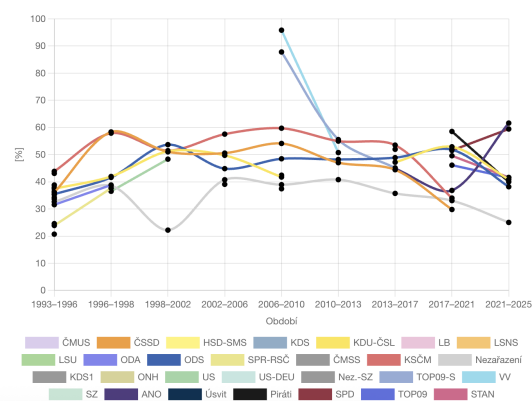
Tento graf znázorňuje, jak se v průběhu jednotlivých období měnila docházka politických stran – ukazuje podíl hlasů označujících „nepřítomen“ nebo „nezúčastnil se“ vůči celkovému počtu členů každé strany, vyjádřený v procentech. **Míra absence závisí na počtu členů v daném politickém subjektu**, a proto může být v některých obdobích vyšší nebo nižší. Vysoká míra absence byla zaznamenána zejména v **prvním období** a také v **osmém období**, pravděpodobně kvůli pandemii COVID-19, kdy absence dosahovala téměř **40%** u několika stran, a u všech politických subjektů byla vyšší než **20%**.



Podobnost hlasování stran v čase (%)

Tento graf ilustruje, jak se během jednotlivých období měnila hlasovací soudržnost mezi politickými stranami. Pokud porovnáme koalice jako **ODS, STAN, a KDU-ČSL** proti stranám jako **SPD, ANO, a ČSSD**, je patrné, že koalice drží pohromadě a častěji se shodují.

● Ano ○ Ne ○ Zdržel se



Obrázek 7.3: Příklad čárového grafu z práce zaměřené na parlamentní hlasování [17]

Tímto způsobem se ukázalo, že komponenty vytvořené v rámci této práce nejsou použitelné pouze pro data měst a krajů, ale lze je efektivně využít i pro jiná veřejně dostupná data. Díky funkcionálnímu přístupu a širokým možnostem konfigurace mohou být použity i pro další analytické nástroje, které budou v systému Zastupko.cz vznikat.

Kapitola 8

Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat webovou knihovnu komponent, které slouží k vizualizaci dat z hlasování zastupitelstev. Zaměřil jsem se na to, aby knihovna byla snadno použitelná, přehledná a zároveň přizpůsobitelná pro různé datové vstupy i potřeby vývojářů. Komponenty měly umožnit jednoduché zobrazení složitých informací, jako je rozložení mandátů, účast, výsledky hlasování nebo mapové vizualizace.

Na začátku jsem se seznámil s tím, jak fungují otevřená data samospráv a co vše se z nich dá zjistit. Prošel jsem různé existující nástroje, které podobná data vizualizují, a snažil se z nich vzít to nejlepší, co by se dalo uplatnit i v mém řešení. Následně jsem analyzoval dostupná data a vizualizace z projektu Zastupko.cz a připravil návrhy komponent, které by tato data co nejlépe prezentovaly. V praktické části jsem se věnoval samotné implementaci. Pro každou vizualizaci jsem vytvořil samostatnou React komponentu, která je konfigurovatelná pomocí vstupních parametrů.

Knihovnu jsem navrhl tak, aby se dala snadno použít a zároveň jednoduše zapojit do stávajícího systému. Všechny komponenty zvládají jak práci s běžnými statickými daty, tak i s daty, která se předávají pomocí funkcí. Hotové vizualizace jsem otestoval přímo v aplikaci Zastupko, kde fungují na reálných datech ze zastupitelstev. Zároveň vizualizace byly prakticky využity v Zastupku pro analýzu a vizualizaci různých dat. Každá komponenta se dá upravit podle potřeby a cílem bylo, aby s ní mohli dobře pracovat jak vývojáři, tak i běžní uživatelé, kteří se chtějí v datech snadno vyznat.

Do budoucna by se knihovna mohla dál rozšiřovat, například o nové typy grafů, víc možností interakce nebo napojení na další datové zdroje. Věřím, že takový nástroj může pomoci zpřehlednit hlasování zastupitelů a přiblížit důležitá rozhodnutí více lidem.

Literatura

- [1] APTIEN. *Jaký je rozdíl mezi daty, informacemi a znalostmi*. 2023. Dostupné z: <https://aptien.com/cs/kb/articles/what-is-difference-data-information>.
- [2] ATTARD, J.; ORLANDI, F.; SCERRI, S. a AUER, S. A systematic review of open government data initiatives. *Government information quarterly*. Elsevier, 2015, sv. 32, č. 4, s. 399–418.
- [3] BARAVALLE, A.; LANFRANCHI, V. a GRIBAUDO, M. Using SVG and XSLT for graphic representation, 2003.
- [4] BEDI, I. *Harnessing Scalable Vector Graphics (SVG) for Effective Data Visualization*. 2023. Dostupné z: <https://hackernoon.com/harnessing-scalable-vector-graphics-svg-for-effective-data-visualization>.
- [5] BOOST LABS. *Static vs Interactive Visualizations – How to Choose* <https://boostlabs.com/static-vs-interactive-visualizations-choose/>. 2024. Online. [cit. 2024-11-22].
- [6] BOSTOCK, M. *What is D3?* 2024. Dostupné z: <https://d3js.org/what-is-d3>.
- [7] BUILT IN. *What Are the 2 Types of Data Visualization?* <https://builtin.com/data-science/data-visualization>. 2024. Online. [cit. 2024-11-22].
- [8] CHART.JS. *Chart.js Documentation*. 2024. Dostupné z: <https://www.chartjs.org/docs/latest/>.
- [9] CHEN, M.; FLORIDI, L. a BORGIO, R. What is visualization really for? *The Philosophy of Information Quality*. Springer, 2014, s. 75–93.
- [10] CODEX, A. C. *Data Visualization with Canvas API: Dynamic Charts and Graphs*. 2024. Dostupné z: <https://reintech.io/blog/data-visualization-canvas-dynamic-charts-graphs>.
- [11] DELIGHT, G. *SVG vs Canvas: A Comparison*. 2023. Dostupné z: <https://medium.com/stackanatomy/svg-vs-canvas-a-comparison-1b58e6c84326>. Accessed: 2025-01-16.
- [12] FEW, S. *Information dashboard design: The effective visual communication of data*. O'Reilly Media, Inc., 2006.

- [13] FREECODECAMP. *How to Use Chart.js for Interactive Data Visualization*. 2024. Dostupné z: <https://www.freecodecamp.org/news/how-to-use-chart-js-for-interactive-data-visualization/>.
- [14] GAO, Y.; JANSSEN, M. a ZHANG, C. Understanding the evolution of open government data research: towards open data sustainability and smartness. *International Review of Administrative Sciences*. SAGE Publications Sage UK: London, England, 2023, sv. 89, č. 1, s. 59–75.
- [15] HIGHCHARTS. *Understanding Highcharts*. 2024. Dostupné z: <https://www.highcharts.com/docs/chart-concepts/understanding-highcharts>.
- [16] IGLESIAS, M. *Pro D3.js*. Springer, 2019.
- [17] LYTVYNYŠYNOVÁ, M. *Rozšíření systému hlasování zastupitelstev o Parlament ČR*. Brno, 2025. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/162921>, [cit. 2025-04-30].
- [18] MAHAJAN, K. N. a GOKHALE, L. A. Comparative study of static and interactive visualization approaches. *International Journal on Computer Science and Engineering (IJCSE)*, e-ISSN, 2018, s. 0975–3397.
- [19] MAREK, T. Efektivní vizualizace dat se zaměřením na základní typy graf. *Dipl. Masarykova Univerzita*, 2014.
- [20] MICROSOFT. *Why Data Visualization is Important*. 2024. Dostupné z: <https://www.microsoft.com/cs-cz/microsoft-365/visio/data-visualization#Why-data-visualization-is-important>. Accessed: 2024-11-21.
- [21] MINISTERSTVO PRO MÍSTNÍ ROZVOJ a MINISTERSTVO VNITRA ČR. *Zpráva o situaci v oblasti otevřených dat ve městech a obcích ČR*. 2017. Dostupné z: <https://opendata.gov.cz/dokumenty:2017-06-28-otevrena-data-ve-mestech>. Dostupné online, citováno 2. dubna 2025.
- [22] MINISTERSTVO VNITRA ČR. *Základní informace pro nově zvolené členy zastupitelstva obce*. 2024. Dostupné z: <https://mv.gov.cz/clanek/zakladni-informace-pro-nove-zvolene-cleny-zastupitelstva-obce.aspx>.
- [23] MISHRA, S. a MISHRA, S. Getting started with highcharts. *Practical Highcharts with Angular: Your Essential Guide to Creating Real-time Dashboards*. Springer, 2020, s. 1–14.
- [24] MUNZNER, T. *Visualization analysis and design*. CRC press, 2014.
- [25] MURRAY RUST, P. Open data in science. *Nature Precedings*. Nature Publishing Group UK London, 2008, s. 1–1.
- [26] ONTOTEXT. *What Is Five-Star Linked Open Data?* 2023. Dostupné z: <https://www.ontotext.com/knowledgehub/fundamentals/five-star-linked-open-data/>.

- [27] SÁNDOROVÁ, D. *Webová aplikace pro analýzu dat z parlamentu České a Slovenské republiky*. Brno, 2025. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/164736>, [cit. 2025-04-30].
- [28] TABLEAU. *Data Visualization Tips For More Effective And Engaging Design* <https://www.tableau.com/visualization/data-visualization-best-practices>. Online. [cit. 2024-11-22].
- [29] TELLER, S. *Data Visualization with d3.js*. Packt Publishing, 2013.
- [30] ZAKLOVÁ, K. *Analýza a vizualizace dat z hlasování Zastupitelstva města Brna*. 2023. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií.
- [31] ÖNCÜLER, U. *Canvas API for Graphics in Web Design*. 2024. Dostupné z: <https://618media.com/en/blog/canvas-api-for-graphics-in-web-design/#core-features-of-the-canvas-api>.
- [32] ČESKO. *Zákon č. 106 ze dne 11. května 1999 o svobodném přístupu k informacím* Sbírka zákonů České republiky, částka 39, s. 2578–2582. 1999. ISSN 1211-1244.
- [33] ČESKO. *Zákon č. 128 ze dne 12. dubna 2000 o obcích (obecní zřízení)* In: Sbírka zákonů České republiky, částka 38, s. 1737–1764. 2000. ISSN 1211-1244.