



Zadání bakalářské práce

Název:	Web lukostřeleckého klubu 1. LK Plzeň 1935
Student:	Jana Schořová
Vedoucí:	Ing. Tomáš Nováček
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Webové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Lukostřelecký klub 1. LK Plzeň 1935 by rád vytvořil nový web pro prezentaci své činnosti s administrační částí pro správu obsahu a aktualit.

Cíl práce:

- 1) Analyzujte aktuální web lukostřeleckého klubu 1. LK Plzeň 1935.
- 2) Analyzujte alespoň tři weby s podobnou tematikou.
- 3) Na základě komunikace s vedením a členy oddílu analyzujte jejich nároky na web po stránce frontendu i backendu.
- 4) Diskutujte možnosti vytvoření webu úplně od základů, s použitím CMS a s použitím headless CMS.
- 5) Navrhněte UI, architekturu a funkce webové aplikace tak, aby splňovala požadavky klienta. UI konzultujte s klientem za pomoci wireframů.
- 6) Aplikaci naimplementujte.
- 7) Aplikaci nasadte a proveďte uživatelské a akceptační testování.



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

Web lukostřeleckého klubu 1. LK Plzeň 1935

Jana Schořová

Katedra softwarového inženýrství
Vedoucí práce: Ing. Tomáš Nováček

12. května 2022

Poděkování

Touto cestou bych ráda poděkovala Ing. Tomášovi Nováčkovi za vedení bakalářské práce, jeho trpělivost a ochotu. Také bych ráda poděkovala vedení 1. lukostřeleckého klubu Plzeň 1935 za nabídku tohoto tématu a jejich čas.

Zároveň bych chtěla poděkovat svým rodičům za podporu během studia, příteli za velkou oporu a povzbudivá slova, díky kterým byla tato práce dokončena, a Jaroslavu Křížovi za pomoc při začátcích studia, které si velice vážím.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principu při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisu, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 12. května 2022

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2022 Jana Schořová. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Schořová, Jana. *Web lukostřeleckého klubu 1. LK Plzeň 1935*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Abstrakt

Tato práce se věnuje tvorbě nového webu pro 1. lukostřelecký klub Plzeň 1935. Výsledný web slouží pro prezentaci aktivit klubu a obsah lze spravovat pomocí administrační části. Práce se zaměřuje na analýzu stávajícího řešení, uživatelských požadavků na nový web a jejich implementaci. Popisuje různé možnosti, jakými lze vytvořit web se správou obsahu. Výsledný web využívá pro administrační část systém Strapi a pro prezentační část staticky generované stránky pomocí Nuxt.js. V závěru práce je popsáno testování a nasazení aplikace.

Klíčová slova webová aplikace, lukostřelecký klub, redesign, systém pro správu obsahu, Strapi, Nuxt.js

Abstract

This thesis focuses on creating a new website for an archery club. The final website serves as a presentation point for the club, its content can be managed with the administrative section of the web. Parts of the thesis focus on analyzing the current website, users requirements for the new one and the new website implementation. It describes various techniques of creating a website using content management systems. Implemented website uses Strapi for the administrative part and Nuxt.js for the presentation part of the website. The end of thesis describes testing and deployment of the new website.

Keywords web application, archery club, redesign, content management system, Strapi, Nuxt.js

Obsah

1	Úvod	1
1.1	Motivace	1
1.2	Cíl práce	1
1.3	Struktura práce	2
2	Analýza	3
2.1	Analýza stávajícího webu	3
2.1.1	Vzhled	3
2.1.2	Technické údaje	6
2.2	Analýza uživatelských požadavků	6
2.2.1	Funkční požadavky	7
2.2.2	Nefunkční požadavky	8
2.3	Rešerše webů s podobnou tematikou	9
2.3.1	Web lukostřeleckého oddílu TJ Opava	9
2.3.2	Web Lukostřelba Brno	10
2.3.3	Web Redhill Archers	11
2.4	Možnosti řešení nového webu	13
2.4.1	Co je CMS?	13
2.4.2	Typy CMS	14
2.4.3	Vlastní řešení systému pro správu obsahu	16
2.5	Možnosti vývoje frontendu	17
2.5.1	HTML	17
2.5.2	CSS	17
2.5.3	CSS preprocesory	17
2.5.4	JavaScript	18
2.5.5	JavaScriptové frameworky	18
2.5.6	Způsoby renderingu	19
2.6	Výběr řešení	21

3	Návrh	23
3.1	Administrační část a backend	23
3.1.1	Strapi	23
3.1.2	Databáze	25
3.2	Prezentační část	25
3.2.1	Nuxt.js	26
3.2.2	Vuex	26
3.2.3	Sass	26
3.2.4	Wireframe	26
3.2.5	Design	32
3.3	Architektura aplikace	32
3.3.1	Architektura MVC	34
3.3.2	Komunikace mezi prezentační částí a backendem	34
4	Implementace	37
4.1	Administrační část a backend	37
4.1.1	Počáteční nastavení	37
4.1.2	Typy obsahu	37
4.1.3	Rozšíření funkcionality	41
4.1.4	Přizpůsobení administrátorského panelu	43
4.2	Prezentační část	43
4.2.1	Konfigurace a souborový systém	43
4.2.2	Vývoj	44
5	Testování a nasazení	53
5.1	Testování	53
5.1.1	Uživatelské testy	53
5.1.2	Akceptační testy	55
5.1.3	Výsledky testování	56
5.2	Nasazení	56
5.2.1	Backend	56
5.2.2	Frontend	57
5.2.3	Webhook	57
5.3	Budoucnost vývoje webové aplikace	57
	Závěr	59
	Bibliografie	61
	A Seznam použitých zkratk	67
	B Obsah příloženého média	69

Seznam obrázků

2.1	Domovská stránka webu 1. LK Plzeň 1935 [1]	4
2.2	Stránka <i>partneři</i> na webu lukostřeleckého oddílu TJ Opava [7] . . .	10
2.3	Domovská stránka webu Lukostřelba Brno [8]	11
2.4	Domovská stránka webu Redhill Archers [9]	12
2.5	FAQ sekce na webu Redhill Archers [9]	12
3.1	Rozdělení typů obsahu na jednorázové a v kolekci	24
3.2	Návrh velkého menu	29
3.3	Návrh stránky <i>Klub</i> : sekce o klubu a lidé v klubu	30
3.4	Návrh stránky <i>Chceš si zastrílet z luku</i> : sekce otázky a odpovědi .	31
3.5	Návrh stránky <i>Chceš si zastrílet z luku</i> : sekce jak se stát členem klubu	31
3.6	Návrh stránky <i>Kontakt</i>	33
4.1	Jednorázové typy obsahu, jejich atributy a komponenty	39
4.2	Typy obsahu v kolekci, jejich atributy a relace mezi nimi	41
4.3	Systém souborů ve složce <i>pages</i>	44
4.4	Komponenta <i>TheHeader</i> a <i>Head</i>	45
4.5	Komponenta <i>HamburgerMenu</i>	46
4.6	Komponenta <i>TheFooter</i>	46
4.7	Sekce <i>Aktuality</i>	47
4.8	Sekce <i>Jak se stát členem klubu</i>	49

Seznam tabulek

3.1	Mapování nového rozložení stránek na stávající rozložení na webu	
	1. LK Plzeň	28

Úvod

1.1 Motivace

Lukostřelba je krásný sport, který však není v České republice příliš rozšířený. Pro lukostřelecké kluby je tedy důležité rozšiřovat povědomí o tomto sportu, a tím získávat nové členy a sponzory. K tomu jim slouží sebe prezentace jak v reálném světě, tak i v tom internetovém. Nejlepší volba na prezentaci nějakých aktivit v klubu jsou nyní sociální sítě, ale důvěryhodné a obecné informace o lukostřelbě a o fungování klubu se primárně vyskytují na webové stránce. Proto je tedy vhodné, aby klub měl vlastní web, na kterém se člověk může dozvědět něco o lukostřelbě, o klubu, kdy probíhají tréninky a jak se vlastně stát členem.

1. lukostřelecký klub Plzeň 1935 má svůj vlastní web, který však nevyhovuje požadavkům vedení klubu. Proto by si představenstvo přálo novou moderní webovou aplikaci, kam by byli trenéři klubu motivováni psát různé aktuality a spravovat obsah, aby veřejnost viděla, že klub funguje a má kvalitní trenéry, kteří mají na starosti střelce různých výkonností.

1.2 Cíl práce

Cílem práce je vytvoření nového webu pro lukostřelecký klub 1. LK Plzeň, který splňuje všechny požadavky vedení klubu a má administrační část, která je uživatelsky přívětivá.

Cílem teoretické části práce je analýza stávajícího webu a požadavků vedení klubu. Dalším cílem je rešerše webů s podobnou tematikou. Posledním cílem teoretické části je rešerše funkčních technologií, které řeší typ webů s administrační částí a jejich porovnání s vlastním řešením.

Cílem praktické části je návrh uživatelského rozhraní, které bude prodiskutováno s vedením klubu pomocí wireframů, návrh architektury a funkcí webové aplikace, aby splňovaly požadavky. Dalším cílem praktické části je implemen-

tace aplikace podle výsledků z teoretické části. Posledním cílem praktické části je nasazení aplikace a otestování uživatelskými a akceptačními testy.

1.3 Struktura práce

Práce je rozdělena do pěti kapitol a závěru. Druhá kapitola popisuje analýzu stávajícího webu 1. lukostřeleckého klubu Plzeň 1935, ve které se zaměřuje na jeho vzhled a technické údaje. Dále je proveden sběr uživatelských požadavků. Následuje rešerše webů s podobnou tematikou, z nichž jsou tři popsány. Poté se kapitola zaměřuje na možnosti řešení nového webu z pohledu administrační a prezentační části. V poslední sekci kapitoly se provede výběr ze zmíněných řešení.

Třetí kapitola se zaměřuje na návrh webové aplikace. Provede se návrh a výběr technologií pro administrační i prezentační část. V prezentační části jsou také popsány wireframy, které byly použity při komunikaci s klientem. Poslední sekce v této kapitole se zaměřuje na architekturu a způsob komunikace mezi prezentační částí a backendem.

Ve čtvrté kapitole je popsána implementace jednotlivých částí webové aplikace. Je zde vysvětlen postup implementace a použité nástroje.

Pátá kapitola popisuje způsob a výsledky testování. Dále vysvětluje nasazení prezentační a administrační části, jaké byly využité platformy a jaké nástroje musely být nastaveny pro správné fungování. Kapitola také popisuje možnou budoucí práci na webové aplikaci.

Analýza

Nynější web [1] byl vytvořen na přelomu roku 2016 a 2017 po domluvě s vedením klubu. Do té doby klub žádné webové stránky neměl, a proto autor musel navrhnout rozložení webu, jaké informace bude nabízet a stavěl tedy řešení tzv. na zelené louce. Svoji tvorbu ale dostatečně nekonzultoval s vedením klubu, a to není z výsledkem spokojeno. Obsah webových stránek chce vedení zachovat, ale design, celkové rozložení stránek a způsob, jakým lze přidávat obsah, není dle jejich požadavků.

V této kapitole se zaměřím na analýzu stávajícího řešení a na požadavky klienta. Dále udělám rešerši webových stránek s podobnou tematikou a tři z nich zanalyzuji. Na konci této kapitoly se podívám na různé možnosti řešení nové webové aplikace.

2.1 Analýza stávajícího webu

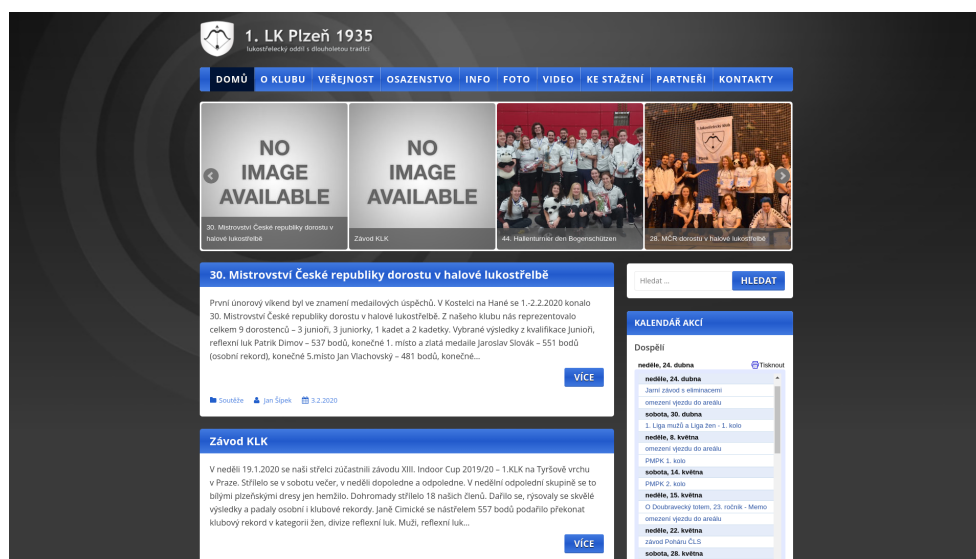
V této sekci se zaměřím na analýzu stávajícího řešení webu 1. lukostřeleckého klubu Plzeň 1935 [1]. Zanalyzuji vzhled webu, jeho rozložení stránek a nedostatky, a poté se podívám na technické údaje webu, tj. na používaný administrační systém a protokol pro přenos dat.

2.1.1 Vzhled

Web je vytvořený na klasické WordPress šabloně a není odlišný od ostatních starších webů, které běží na této šabloně. Není tedy ničím unikátní a nezaujme. Má zastaralý a neoriginální design, který v dnešní době již není žádoucí. Design stávajícího webu je z velké části responzivní. V jednom rozlišení jen odskočí poslední záložka navigace na druhý řádek.

Web je rozdělen do tří hlavních částí. První částí je hlavička, ve které se nachází logo, navigace a slideshow. Druhá část je hlavní část stránky, která se skládá z postranního panelu a obsahu každé stránky. Na obrázku 2.1 je

2. ANALÝZA



Obrázek 2.1: Domovská stránka webu 1. LK Plzeň 1935 [1]

snímek domovské stránky, kde je vidět hlavička a část postranního panelu a těla stránky. Poslední částí webu je patička.

2.1.1.1 Hlavička

Hlavička je složena ze tří komponent a zůstává na každé stránce stejná. První z komponent obsahuje obrázek, název klubu a text. Celá tato komponenta obsahuje odkaz na domovskou stránku. Obrázek, který je na stránce jako logo, vypadá jinak než oficiální logo klubu. Text, který je k obrázku přidán, obsahuje název klubu a text ve znění „lukostřelecký oddíl s dlouholetou tradicí“. Dle vedení klubu se nemá klub pojmenovávat jako oddíl, proto by tento text na webové stránce být neměl. Celá kombinace obrázku a textu je v jediném obrázku ve formátu PNG.

Druhá komponenta je navigace. Navigace webu obsahuje deset položek:

- domů,
- o klubu,
- veřejnost,
- osazenstvo,
- info,
- foto,
- video,
- ke stažení,
- partneři,
- kontakty.

Položky *veřejnost* a *info* mají nejasný obsah. Samotné pojmenování záložky *veřejnost* je dle mnohých členů klubu nešťastné. V záložce se nachází

informace o akcích pro veřejnost na střelnici, informace o firemních akcích, které klub může pořádat, postup, jak se stát členem klubu, a informace o kurzech nováčků. Jsou to tedy informace pro návštěvníka webu, jakými způsoby si může zkusit lukostřelbu.

Druhá nejasná položka *info* obsahuje mapu ukazující na střelnici a mapu, která značí cestu, jak se dostat do tělocvičny, kde se střílí v zimních měsících. Dále obsahuje tréninkové časy, odkaz na informace, jakým způsobem se stát členem klubu, informace o soustředění a finanční stránku sportu, která je skrytá dole v přiloženém dokumentu. Mapy jsou i v sekci *kontakty*, a tak se jedná o duplicitní informaci. Finanční stránka sportu by měla být u informací, jak se stát členem klubu, protože i finance jsou jedna z věcí, která návštěvníka webu zajímá. Sekce *info* by v tomto případě mohla být jen o tréninkových časech a informacích o soustředění.

Z položek v navigaci by mohly být položky *foto* a *video* sloučené dohromady, protože ani jedna stránka nemá moc obsahu. Navigace není po sescrollování vidět. Na zařízeních s šířkou 1025 obrazových bodů a více se zobrazí šipka v pravém dolním rohu, která odkazuje na vrchol webové stránky. Navigace je na zařízeních pod 1025 obrazových bodů formou hamburger menu. Také zůstává po sescrollování mimo obrazovku, ale šipka odkazující na vrchol webové stránky tam není, proto se musí uživatel pro další použití navigace přesunout na začátek webové stránky pomocí scrollování, a to není uživatelsky přívětivé.

Poslední komponenta v hlavičce je slideshow aktualit. Každý slide je ve formě obrázku k dané aktualitě a nadpisem aktuality, který překrývá spodní stranu obrázku. Když není nahraný žádný obrázek k aktualitě, v této slideshow se aktualita objeví s výchozím obrázkem, který má formu šedého obrázku s nápisem „no image available”. To ale není pěkné v rámci prezentace činnosti klubu, protože je to aktuálně první věc, která se objeví na hlavní stránce.

Aktuality se ve slideshow posouvají automaticky po pěti sekundách, šipkami po stranách nebo posunutím při držení levého tlačítka myši či prstem na dotykových obrazovkách. Použitím poslední možnosti se aktuality vždy posunou jen o jednu aktualitu neohledně na to, jak daleko uživatel slideshow posunul. Uživatel může na každou aktualitu kliknout, ale v tomto případě nastává problém. Kliknutí a následné přesměrování na danou aktualitu fungovalo ze tří testovaných webových prohlížečů pouze na jednom. Na ostatních prohlížečích funguje prokliknutí do aktuality, jen když se celá slideshow pohne jedním z uvedených způsobů a v animaci pohnutí uživatel klikne na některou z aktualit. Když uživatel kliká na nějakou z aktivit ve statickém stavu, celá slideshow se posouvá o pár obrazových bodů doleva.

V hlavičce je pod třetí komponentou schovaný vyhledávací formulář, který je kvůli přidanému CSS skrytý. Má průhlednost nastavenou na nulovou hodnotu a viditelnost na skrytý, a proto nevidím důvod, proč tento prvek je v hlavičce. Prvek na vyhledávání se navíc nachází v jiné části webové stránky.

2.1.1.2 Tělo stránky

Druhá část webu je rozdělena do dvou hlavních sekcí. První sekce je samotný obsah každé stránky a druhá je postranní panel, který obsahuje pět komponent. Nachází se zde výše zmíněné textové pole na vyhledávání. Dále obsahuje sekci se třemi kalendáři. Kalendáře jsou vloženy do `iframe`, který byl vygenerován v aplikaci Google Kalendář [2]. Každý kalendář je pro jinou věkovou kategorii. Po kalendářích následuje komponenta s odkazy na rubriky. První odkaz je na fotogalerii, která se zobrazí jen jako nadpisy alb a až po kliknutí se zobrazí fotografie. Další dva odkazy zobrazí aktuality podle zvoleného tématu. V postranním panelu je také komponenta s názvem *Odkazy*, kde je jediný odkaz na lukostřeleckou databázi závodů a výsledků. Poslední komponenta v postranním panelu jsou partneři lukostřeleckého klubu, kde jsou loga hlavních partnerů. Jsou to jen obrázky v různých bitmapových formátech a neobsahují odkazy například na webové stránky daného partnera.

2.1.1.3 Patička

Patička je poslední část webu a nemá moc obsahu. Obsahuje jen značku copyright, rok 2017, kdy byla patička nejspíše posledně aktualizována, a email na administrátora stránky. Neobsahuje například odkazy na různé sociální sítě lukostřeleckého klubu, které jsou v patičce běžné.

2.1.2 Technické údaje

Jak již bylo řečeno, web byl děláný ve WordPressu, což je tradiční systém pro správu obsahu. Více o něm v sekci 2.4.2.1. Použitá verze WordPressu je verze 4.7., která vyšla na konci roku 2016. Od té doby do něj přibylo hodně vylepšení, které usnadňují práci s administračním panelem, například nový editor textu nebo nové widgety pro obrázky, video a audio, a které tím pádem v současné administrační části chybí. Protože WordPress neměl do konce roku 2016 podporu protokolu HTTPS, web běží jen na protokolu HTTP, ale to není problém, protože na webu se nevyplňují žádná data, která by musela být šifrována. WordPress používá jazyk PHP a jeho verze na tomto webu je verze 5.6. Protože nemám přístup k souborům aplikace a server nemá nastavenou databázi, nemohu zjistit, zda a jakou používá webová aplikace databázi.

2.2 Analýza uživatelských požadavků

Web lukostřeleckého klubu slouží k prezentaci klubu, jeho aktivit, také jako informační kanál pro osoby, které by chtěli začít s lukostřelbou, a pro členy klubu, kteří se chtějí dozvědět o nadcházejících závodech. Klientovi se obsah na stávajícím webu líbí a chtěl by ho zachovat a doplnit. Shodli jsme se, že obsah by bylo třeba také přeorganizovat.

Nový web má primárně sloužit k prezentaci klubu a jeho aktivit. Musí být vidět, že klub funguje a střelci mají dobré zázemí, ze kterého se odvíjí i soutěžní úspěchy. Web naopak nemá sloužit jako reklama pro nabírání nových členů, protože v současné chvíli klub nové členy nenabírá. Avšak tato skutečnost se může v budoucnu změnit, proto je třeba informace o nabírání členů zachovat.

Web má možnost správy obsahu, kterou klient využívá. Cílem nového webu je udělat správu obsahu přijatelnější pro uživatele tak, aby věděli, jak se systémem pracovat, a aby je složitá administrace neodrazovala od psaní.

Z našeho rozhovoru s klientem [3] jsem dala dohromady seznam funkčních a nefunkčních požadavků. Cílem nového webu je řešení všech těchto funkcionalit a splnění nároků na web.

2.2.1 Funkční požadavky

Funkční požadavky definují funkcionality, které by měla výsledná aplikace obsahovat. Většina funkčních požadavků se týká administrační části.

F1 – Role uživatelů v administrační části

Systém bude podporovat přiřazení konkrétní role uživatelům, kteří budou mít přístup do administrační části. Role budou tři: administrátor, správce obsahu a správce aktualit. Tyto role bude moci přiřadit přihlášený administrátor v administrační části.

F2 – Správa obsahu

Aplikace bude podporovat přidávání, upravování a mazání obsahu na klientem určených stránkách. Obsah budou moci spravovat jen přihlášení uživatelé v administrační části s rolí správce obsahu.

F3 – Správa aktualit

Přihlášení uživatelé s rolí správce aktualit v administrační části budou moci přidávat, upravovat a mazat aktuality. Aktualita bude obsahovat název, den vydání, jméno autora, obsah aktuality, titulní fotografii a dodatečné fotografie. Dále některé aktuality obsahují výsledky ze závodů, které budou moci být přidávány pomocí souboru typu CSV.

F4 – Tvorba profilů

Aplikace bude podporovat tvorbu profilů pro osoby v klubu. Profil může být přiřazen ke konkrétní osobě ve vytvořených seznámech. Osoba bude moci mít více profilů v závislosti na její roli v klubu. Profily bude moci vytvářet uživatel přihlášený v administrační části s rolí správce obsahu.

F5 – Načítání dat ze souboru

Aplikace bude umožňovat načítat data ze souboru pro následující datové modely v administrační části: členové klubu, úspěchy, klubové rekordy a již zmíněné výsledky v aktualitě. Soubory budou typu CSV. Data se po načtení budou moci upravovat.

F6 – Vypisování termínů kurzu pro nováčky

V administrační části budou moci přihlášení uživatelé vypisovat termíny na kurz lukostřelby pro začátečníky. Tyto termíny se pak objeví na webové prezentaci ve formuláři.

F7 – Kontaktní formuláře

Webová prezentace bude obsahovat tři formuláře (lukostřelba pro veřejnost, nováčkovský kurz, kontaktní formulář). Po vyplnění formuláře se data odešlou na přiřazený e-mail a pošle se automatická odpověď na e-mailovou adresu ve formuláři. Data se nemusí ukládat.

2.2.2 Nefunkční požadavky

Nefunkční požadavky jsou všechny požadavky, které nesouvisejí s funkčním aspektem softwaru. Definují technické nároky kladené na aplikaci.

N1 – Rozdělení na administrační a prezentační část

Aplikace se bude skládat ze dvou částí. První část bude administrační, kde budou moci přihlášení uživatelé vykonávat funkční požadavky F1–F6.

Druhá část bude webová prezentace klubu. Bude obsahovat informace, které jsou na stávajícím webu, a doplňující informace, které na současném webu chybí (osobnosti, úspěšnost).

N2 – Responzivita

Prezentační část bude responzivní. Bude podporovat zobrazení na mobilní zařízení, tablet a desktop. Administrační část bude podporovat zobrazení zatím jen na desktop.

N3 – Podpora prohlížečů

Aplikace bude podporovat následující verze prohlížečů (jedná se o čtyři nej-používanější prohlížeče v Evropě [4]):

- Google Chrome verze 88 a novější a mobilní verze Chrome for Android,
- Mozilla Firefox verze 97 a novější,

- Safari verze 15.1 a novější a mobilní verze Safari on iOS,
- Microsoft Edge verze 98 a novější.

N4 – Počet uživatelů v aplikaci

Do administrační části budou mít přístup jen vybraní lidé z řad představenstva a trenérů. Jde o velmi malý počet uživatelů, proto se nemusí řešit vysoké požadavky na zátěž systému.

2.3 Rešerše webů s podobnou tematikou

V České republice v současné době existuje 65 lukostřeleckých klubů, z nichž dvě třetiny mají nějakou webovou stránku. Většina z těchto stránek má rozložení obsahu a design podobný jako má současná webová stránka 1. lukostřeleckého klubu Plzeň 1935, v této sekci dále jen „web LK Plzeň“.

Po prozkoumání zahraničních lukostřeleckých klubů a jejich webových stránek jsem zjistila, že velká část z nich je postavena také na podobném designu.

Mezi všemi jsem vybrala webové stránky dvou klubů v České republice, tj. Lukostřelba Opava a Lukostřelba Brno, a jednu stránku zahraničního klubu Redhill Archers. Každý z webů má nějaký zajímavý nápad, který bych následně chtěla implementovat ve své práci.

2.3.1 Web lukostřeleckého oddílu TJ Opava

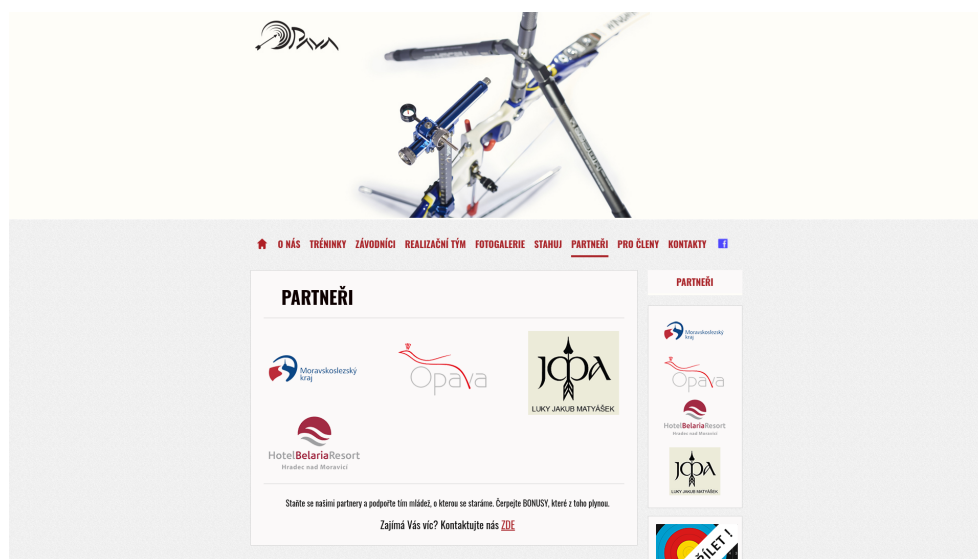
Tato webová stránka [5] informuje o lukostřeleckém oddílu TJ Opava. Vzhledem i rozložením obsahu je velmi podobná webu LK Plzeň, ale není responzivní. Jak je vidět na obrázku 2.2, obsahuje horizontální menu a dlouhý postranní panel s informacemi.

V postranním panelu jsou loga partnerů i s odkazy na jejich webové stránky, odkaz na informace pro zájemce, výpis událostí z minulé sezóny, časy tréninků a odkaz na jejich detailní rozpis. Poslední položka je rychlý kontakt, tj. email a telefonní číslo, u kterého není uvedeno, komu patří. Postranní panel je velmi dlouhý a většina stránek má kratší obsah, proto pod obsahem vzniká dlouhé prázdné místo.

Menu na webu obsahuje 11 položek. Zaujala mě položka *realizační tým*, kde jsou vypsáni trenéři a rozhodčí i s fotografiemi. Podrobnější informace jsou napsány jen u hlavního trenéra. Tato stránka je cenná ve chvíli, kdy se návštěvníci webu chtějí dozvědět, jací lidé trénují sportovce v klubu. Na webu LK Plzeň je podobná stránka a ta bude na novém webu zachována.

Další položka v menu jsou *závodníci*. Zde jsou podobně jako na stránce *realizační tým* vypsáni střelci a jejich fotografie. Karty střelců vedou odkazem na jejich profil na stránce Rcherz [6], kam se píšou výsledky závodů. Nemí

2. ANALÝZA



Obrázek 2.2: Stránka *partneři* na webu lukostřeleckého oddílu TJ Opava [7]

jasné, zda vypsání střelci jsou všichni v klubu, nebo je to jen vybraná skupina. Tato sekce musí podléhat neustálé správě, protože střelci se kdykoliv mohou rozhodnout skončit s lukostřelbou.

Dále mě zaujala stránka *partneři*, která je zobrazena na obrázku 2.2. Na stránce jsou loga partnerů spolu s odkazy na jejich webové stránky. Pod logy je motivační nápis, proč by se nějaká společnost měla stát partnerem klubu. Podobný styl motivace využijí i na novém webu, protože má být zaměřen právě na získání nových partnerů či sponzorů.

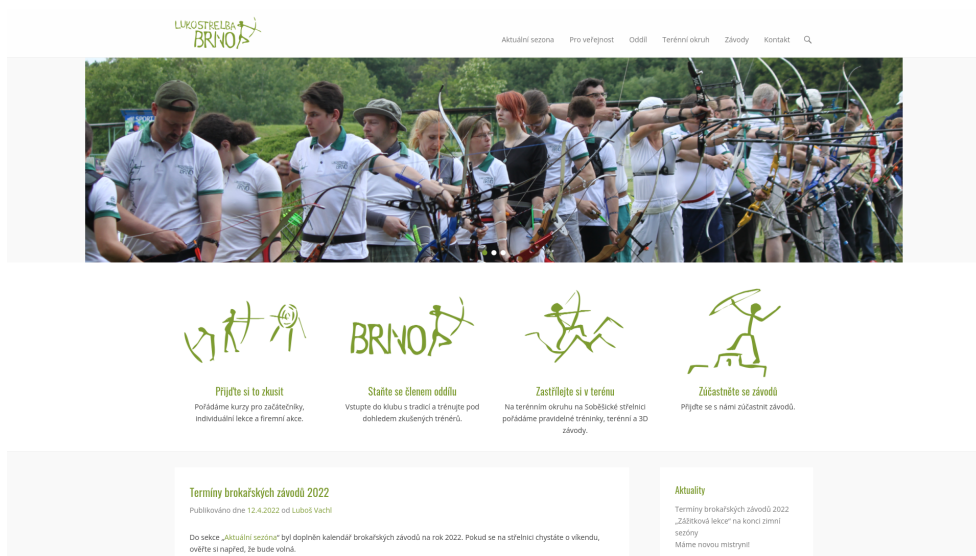
Domovská stránka webu, stejně jako web LK Plzeň, obsahuje jen výpis aktualit. Poslední aktualita je z roku 2020, takže to na první pohled vypadá, že klub není aktivní.

2.3.2 Web Lukostřelba Brno

Druhý vybraný web [8] je podle kontaktů v patičce lukostřeleckého oddílu TJ Start Brno a lukostřeleckého klubu Brno 05, ale obsah na webu je tvořený jen pro jeden subjekt. Není jasné, k jakému subjektu zmíněné informace patří. Web je vytvořen pomocí WordPressu a oproti předchozímu webu je až na nějaké drobnosti responzivní.

Jak je vidět na obrázku 2.3, má stejně jako předchozí zmíněný web horizontální menu, obsahovou část a postranní panel. V postranním panelu je nepřehledný výpis tří posledních aktualit a přihlašovací formulář, který má pod tlačítkem na přihlášení vypsáno PHP varování spojené s použitým widgetem. Dále je v postranním panelu také doplňující menu k aktuální sekci.

2.3. Rešerše webů s podobnou tématikou



Obrázek 2.3: Domovská stránka webu Lukostřelba Brno [8]

Na hlavní stránce obsahuje obrázkovou navigaci (obrázek 2.3), která má pomoci se lépe zorientovat lidem, co chtějí začít s lukostřelbou. Tento způsob navigace je zajímavý a jeho funkce na předchozím webu a na webu LK Plzeň chybí.

Na webu jsou aktualizované informace týkající se aktuální sezóny, ale také se často objevují starší data (například u informacím k náborům), které nedělají dobrý dojem. Některé důležité informace (například kolikrát týdně mohou členové klubu střílet), které by měly být mezi informacemi o klubu, jsou skryty v jedné z aktualit. V této aktualitě jsou popsány odpovědi na otázky, co by si člověk mohl položit, když se rozhodne dělat tento sport. Tato forma otázek a odpovědí je velmi užitečná a společně s klientem jsme se ji rozhodli implementovat na nový web.

Web je velmi přehledný a designem minimalistický. Ve velké části webu se nevyskytují žádné obrázky. Návštěvník webu si tak nemůže představit, kdo v klubu trénuje, jací lidé tam střílí či jak vypadá střelnice.

2.3.3 Web Redhill Archers

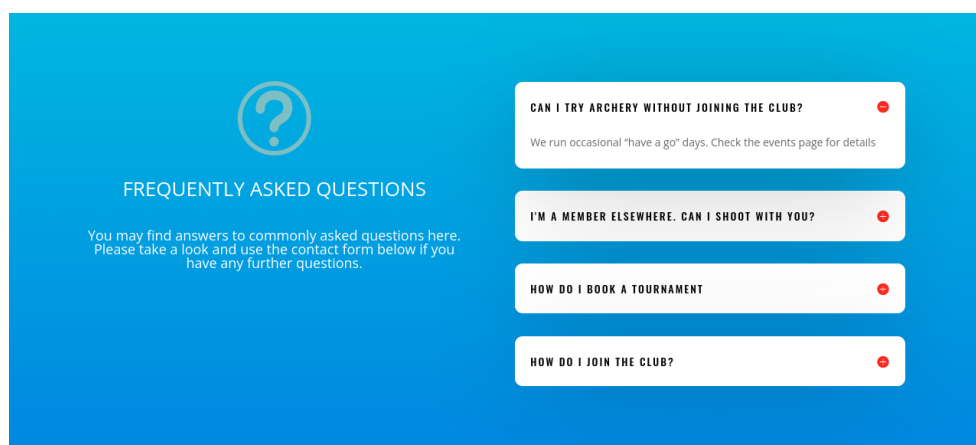
Tento web [9] patří lukostřeleckému klubu Redhill Archers, který sídlí v Birminghamu v Anglii. Web je velmi jednoduchý a v moderním stylu. Je až na některé části responzivní a většina částí využívá animované přechody. Klub nemá nikde na webu vlastní logo.

Jak je vidět na obrázku 2.4, v úvodu stránky je obrázek, na němž je název a další informace o klubu. Název v klubu v některých rozlišeních zmizí. Menu má formu rozklikávacího vertikálního menu. Jeho ikona na rozkliknutí zůstává

2. ANALÝZA



Obrázek 2.4: Domovská stránka webu Redhill Archers [9]



Obrázek 2.5: FAQ sekce na webu Redhill Archers [9]

při scrollování v horní části obrazovky, ale je nevýrazná na pozadí, které se na stránce objevují. Některé položky menu odkazují na nějakou sekci na domovské stránce a některé na úplně jinou stránku. Pokud jsem na jiné stránce a kliknu na položku, která odkazuje na sekci na domovské stránce, otevře se domovská stránka nahore a až po chvíli se automaticky posune na danou sekci. Tato funkcionality je matoucí.

Menu obsahuje sedm položek, z nichž dvě položky *events* a *tournaments* vedou na stejnou stránku. Dále je v menu matoucí *FAQ* položka. Pokud jsem na domovské stránce nebo na stránce *beginners*, po kliknutí na *FAQ* položku v menu se stránka posune na sekci *FAQ* v domovské stránce. Pokud jsem však na stránce *events*, tato položka posune na sekci *FAQ* na těchto stránkách, která má jiný obsah než na domovské stránce. Sekce *FAQ* (obrázek 2.5) je zajímavá a s kombinací odpovědí na otázky na předešlém webu by se mohla dostat i na nový web.

Domovská stránka obsahuje informace o klubu, odkaz na informace pro začátečníky a tři karty s informacemi o činnostech klubu. Dvě karty jsou jako odkaz na jinou stránku, poslední je jen informační. Navíc v poslední kartě není ve větších rozlišeních vidět poslední řádek textu. Dále následuje sekce s často položenými otázkami. Otázky se dají rozbalit pro získání odpovědi. Tato sekce je zajímavá a je to další způsob, jak udělat sekci s otázkami, kterou budu implementovat na nový web. Poslední část obsahuje kontaktní formulář, mapu, kde se nachází klub, a sekci, kde jsou loga různých organizací podporujících lukostřelbu ve Velké Británii. Web neobsahuje patičku.

Některé sekce mají bílé písmo na barevném pozadí, které při menších rozlišení přesahuje mimo sekci a není na bílém pozadí, které je okolo, vidět. Také se na nějakých stránkách objevuje ten samý nadpis sekce dvakrát.

2.4 Možnosti řešení nového webu

V této sekci prodiskutuji různé možnosti, jak vytvořit web se systémem pro správu obsahu. Zaměřím se konkrétně na dva typy existujících řešení a na možnost napsat vlastní řešení.

Jak bylo zmíněno v prvním nefunkčním požadavku klienta, aplikace se bude skládat ze dvou částí. První část bude webová prezentace a druhá část bude část administrační, kde budou moci přihlášení uživatelé např. spravovat obsah a přidávat aktuality. Pro tento typ webových aplikací existují řešení v podobě CMS.

2.4.1 Co je CMS?

Systém pro správu obsahu (CMS z anglického content management system) je software, který poskytuje efektivní správu a publikaci obsahu. Obsah je typicky uložen v databázi a publikován v nějaké prezentační formě například na webovou stránku. CMS umožňuje spolupráci více uživatelů při přidávání,

úpravě, formátování a publikaci obsahu. Uživatelé nepotřebují ke správě obsahu žádné technické znalosti ohledně webového vývoje a lze jim v CMS přiřadit práva, co smí a nesmí spravovat. [10, 11]

2.4.2 Typy CMS

Web se systémem pro správu obsahu je rozdělen do dvou částí. První částí je tzv. backend, který má za úkol administraci. V backendu mohou přihlášení správci vytvářet a upravovat obsah. Druhá část je tzv. frontend, který vytvořený obsah zobrazuje.

V posledních letech se začaly rozšiřovat různé architektury CMS. Každá architektura řeší komunikaci backendu a frontendu jinak, hodí se na různá řešení webu a má své výhody a nevýhody. Vybrala jsem dvě architektury, které popíšu a porovnáám je s vlastním řešením systému pro správu obsahu.

2.4.2.1 Tradiční (coupled) CMS

Coupled (nebo také tzv. tradiční) CMS spojuje dohromady část se správou obsahu i část, kterou vidí veřejnost. Vše běží na jednom systému. Frontend, backend, databáze a veškeré dodatečné soubory jsou uloženy na stejném úložišti. Vytvářený obsah je úzce spojený s tím, jak se zobrazí na webové stránce. Pro přidání více funkcí na web lze využít pluginy, které jednotlivé coupled CMS nabízí. Tradiční CMS se hodí na jednoduché weby, kdy klient nepotřebuje obsah na jiných platformách. [12, 13, 14]

Mezi výhody tradičního CMS [15, 16] patří:

- Web lze pomocí coupled CMS rychle vytvořit a nastavit.
- Dokáže ho zprovoznit i člověk s minimálními znalostmi programování.
- Pro frontend se může využít předem daná šablona nebo si lze vytvořit vlastní.
- Vše lze spravovat z jednoho systému.
- Pro základní využití je většina CMS bez poplatků.

Nevýhody tradičního CMS [15, 16] jsou:

- Frontend se nedá maximálně přizpůsobovat.
- Protože jsou části systému (frontend a backend) spjaty, dělat změny na jedné části znamená změnit celý systém. Nelze jednoduše vyměnit frontend nebo backend.
- Obsah nelze využít pro jiné platformy (lze ho využít pouze pro web).

V posledních letech se tradiční CMS snaží přidávat funkce i pro použití jako headless CMS. Příklady coupled CMS jsou WordPress, Drupal a Joomla!, které jsou popsány v následujícím seznamu:

WordPress [17] je nejrozšířenější [18] ze všech dostupných CMS. Jedná se o open-source software, který používá jazyk PHP a databázi MySQL. Díky velkému rozšíření existuje mnoho návodů (i v českém jazyce), jak si WordPress nastavit a spustit na vlastním serveru. Původně byl určen na tvorbu blogů, ale postupem času rozšířil působnost i v tvorbě jednoduchých webů a eshopů. Má velký výběr šablon a pluginů, díky kterým si ho lze přizpůsobit. U pluginů se musí dávat pozor na jejich aktuálnost, protože se často stávají terčem hackerských útoků. [19, 20, 21]

Drupal [22] je open-source CMS, které nabízí velké přizpůsobení systému. Oproti WordPressu nenabízí předpřipravené řešení, ale jen kostru, kterou si uživatel přizpůsobí dle svého uvážení. K tomu uživatel potřebuje určité znalosti programování, proto Drupal není vhodný pro úplné začátečníky. Prioritou Drupalu je vysoká přizpůsobivost a bezpečnost. Používá na větší korporátní weby. [19, 20, 21]

Joomla! [23] stojí někde mezi WordPressem a Drupalem. Kombinuje vysokou přizpůsobivost Drupalu s jednoduchou konfigurací WordPressu. Pro přizpůsobení lze využít mnoho šablon a rozšíření, ale spousta z nich je placených. Instalace rozšíření je pro uživatele s minimálními technickými znalostmi poměrně těžká. [19, 20, 21]

2.4.2.2 Headless CMS

Oproti tradičnímu CMS nemá headless CMS frontend. Má tedy jen backendovou část na správu obsahu, který může být distribuován např. pomocí API. Umožňuje vývojářům vytvořit jedinečný frontend na jakékoli platformě (web, mobilní aplikace, IoT atd.). [15, 24, 25]

Výhody headless CMS [15, 16] jsou:

- Frontend se může tvořit jakýmkoliv způsobem. Tím je zaručena flexibilita ve výběru technologií.
- Obsah může být distribuován na kteroukoliv platformu.
- Případná změna ve frontendu nebo v backendu nezasáhne celý systém, ale jen danou část, které se změna týká.
- Lepší škálovatelnost, reaguje na nové technologie a nástroje.

Mezi nevýhody headless CMS [15, 16] patří:

- Není definovaná prezentační vrstva, a proto jsou potřeba dodatečné technologie na její zprostředkování.

2. ANALÝZA

- Uživatel si nemůže prohlédnout, jak bude obsah vypadat na webu, dokud není publikovaný.
- Na vývoj frontendu jsou potřeba vývojáři.

Mezi headless CMS patří systémy Strapi, Contentful a Netlify CMS, které jsou popsány v následujícím seznamu:

Contentful [26] je cloud-based CMS, které je hostováno samotnou službou. Nelze si tedy Contentful stáhnout a hostovat na svém serveru. Specializuje se na samotný obsah a backend, proto obsahuje nástroje na tvorbu vlastního modelu obsahu a schémat. Umožňuje integraci s ostatními službami, jako jsou Dropbox a Google Analytics. Na distribuování obsahu využívá RESTful API a nyní i GraphQL. [27, 28]

Strapi [29] je open-source headless CMS, které je postaveno na Node.js. Výběr databázového stroje a hostingu nechává na uživateli. Podporuje RESTful i GraphQL API na distribuci obsahu. Uživatel má možnost si přizpůsobit i vizuální stránku administrační části. [27, 30]

Netlify CMS [31] je open-source headless CMS, které může být použito s jakýmkoliv generátorem statického webu (SSG, více v sekci 2.5.6.3). Je postaveno jako React aplikace. Na rozdíl od výše uvedených CMS nepoužívá k distribuci dat API, ale má obsah uložený na Git (verzovací systém) vedle vlastního kódu frontendu. Díky tomu je obsah dobře verzovatelný a změny obsahu se dají dělat přímo v Gitu. [27, 32]

2.4.3 Vlastní řešení systému pro správu obsahu

Pro vlastní řešení webu se systémem pro správu obsahu je třeba vytvořit zvlášť administrační část a prezentační část. Obě tyto části komunikují s backendem, který ukládá data do databáze.

U administrační části se musí řešit zabezpečení přístupu do aplikace a přiřazení role pro každého uživatele. Dále je nutné zprostředkovat uživatelsky přívětivý editor textu a výstup správně ukládat do databáze. V neposlední řadě je také nezbytné navrhnout a implementovat vhodné ukládání a distribuci obrázků, které jsou velkou součástí prezentační části.

Stejně jako v architektuře headless CMS je nutné vytvořit prezentační část. Tato část zobrazuje v uživatelsky přívětivé formě data, která získá z backendu. Výhody vlastního řešení jsou:

- obě části webu vytvořené přesně na míru požadavků uživatele,
- oproti komerčním CMS žádné zbytečné funkcionality v administrační části,
- absolutní svoboda v použitých technologiích.

Nevýhody vlastního řešení jsou:

- velká časová náročnost vytvoření vlastního řešení,
- po nasazení do produkce povinnost údržby webu,
- náchylnost na bezpečnostní chyby.

Dříve se vyvíjely vlastní systémy pro správu obsahu také z důvodu, že tradiční CMS je v některých ohledech velmi limitující. Headless CMS však dává svobodu ve vývoji frontendu a možnost přizpůsobení administrační části. Vlastní řešení se tak hodí už jen pro velmi specifickou skupinu požadavků. [33]

2.5 Možnosti vývoje frontendu

Pro headless CMS i pro vlastní řešení je nutné vytvořit prezentační část. Běžné technologie pro vývoj frontendu jsou HTML, CSS a JavaScript. K urychlení vývoje se nyní využívají nástroje jako jsou například CSS preprocesory nebo JavaScriptové frameworky. Je ale možnost tyto nástroje nevyužít a veškerý kód na frontendu si napsat ručně.

2.5.1 HTML

HTML neboli Hypertext Markup Language je základní stavební kámen webu. Definuje význam a strukturu obsahu. Byl postaven na standardu SGML (Standard Generalized Markup Language). Jazyk je tvořen elementy a jejich atributy. Poslední verzí je HTML 5. [34]

2.5.2 CSS

CSS neboli Cascading Style Sheet je jazyk stylů, který popisuje, jak se mají prvky v HTML (a v XML včetně dialektů) zobrazovat na určitém médiu, například na obrazovce nebo na papíře. Díky němu můžeme oddělit strukturu dat od vizuální prezentace. CSS patří mezi základní jazyky otevřeného webu a je standardizován napříč prohlížeči podle specifikace W3C (The World Wide Web Consortium, více v [35]).

Styly se definují pomocí sady pravidel. Jedno pravidlo se skládá ze selektoru a bloku deklarácí. Každá deklaráce se skládá z vlastnosti a hodnoty a jsou od sebe oddělené středníkem. [36]

2.5.3 CSS preprocesory

CSS preprocesory zjednodušují práci se styly, protože přidávají nástroje (například vnořené selektory nebo dědění), které v čistém CSS neexistují. Díky nim je CSS více čitelné a snadněji udržitelné. Preprocesory mají svoji unikátní

syntaxi, ze které generují CSS. Existuje mnoho CSS preprocesorů, nejvíce používanými jsou Sass a LESS.

Pro použití preprocesorů je nutné mít na serveru kompilátor CSS, nebo existuje možnost soubor zkompileovat ve vývojovém prostředí a následně ho zkompileovaný nahrát na server. [37]

2.5.4 JavaScript

JavaScript je interpretovaný, objektově orientovaný a multiplatformní jazyk, který je především známý jako skriptovací jazyk pro webové stránky. Umožňuje implementovat složitější funkce. Díky němu lze zobrazovat na stránkách například dynamický obsah, interaktivní mapy nebo animovanou grafiku. Běží na klientské části webu, což může být použito při programování toho, jak se má web chovat při výskytu události, například při kliknutí na tlačítko. K tomu se používají API jako je například DOM API, které slouží k manipulaci s HTML a CSS. Díky němu lze na stránce měnit a odstraňovat HTML prvky nebo dynamicky měnit styly. Dalším příkladem je API pro získání dat ze serveru. [38]

Standardy jazyka jsou ECMAScript Language Specification (ECMA-262) a ECMAScript Internationalization API specification (ECMA-402).

2.5.5 JavaScriptové frameworky

Frameworky v JavaScriptu výrazně usnadňují tvorbu dynamických a interaktivních aplikací a jsou nedílnou součástí tvorby moderního webu. Díky nim jsou aplikace lépe škálovatelné a udržovatelné. Využívají předepsaného a znovupoužitelného kódu, který řeší běžné úkoly jako například práce s DOM API. Mají předepsanou kostru, kterou může vývojář rozšířit nebo pozměnit. Mezi nejpopulárnější frameworky patří React, Angular a Vue.js. [39]

2.5.5.1 React

React [40] byl vydán firmou Meta v roce 2013 a v podstatě se nejedná o framework. Je to knihovna pro renderování UI komponent. Pro vytvoření aplikace je používán v kombinaci s dalšími knihovnamy jako je ReactDOM (pro webové stránky) a React Native (pro mobilní aplikace). První zmíněná kombinace je velmi často používaná, a proto je React považován za framework.

React využívá speciální JSX funkci, která umožňuje vývojářům použít HTML pro aktualizaci DOM a vytvořit virtuální DOM pro vylepšení výkonu. Používá také znovupoužitelné komponenty organizované ve stromové struktuře a funkci jednosměrné vazby (one-way binding), což znamená, že informace je přenášena pouze v jednom směru, a to z nadřazené komponenty do podřazené komponenty.

Je jednoduchý na naučení a rychlý na vývoj díky znovupoužitým komponentám. Má velkou komunitu a podporu společnosti Meta. Nevýhodou je nedůkladná dokumentace. [41, 42]

2.5.5.2 Angular

Angular [43] je open-source framework, který vyšel roku 2010 pod firmou Google. Používá jazyk TypeScript. Je založen na tvorbě komponent a využívá deklarativní šablony HTML. Jeho hlavní předností je funkce obousměrné vazby (two-way binding). Tato funkce umožňuje sdílení všech změn v nadřazené komponentě s podřízenou komponentou a obráceně, což zmenšuje prostor pro chyby.

Framework lze využít na více platformách, takže použití není limitováno pouze na webový prohlížeč. Má velmi velkou komunitu a podporu společnosti Google.

Nevýhodou tohoto frameworku je, že je velmi komplexní a nehodí se na menší aplikace. Je také obtížné se s ním naučit pracovat. [41, 42]

2.5.5.3 Vue.js

Framework Vue.js [44] vyšel v roce 2014 a zkombinoval nejlepší funkce z existujících frameworků. Je velmi snadný a jednoduchý na naučení. Stejně jako Angular rozšiřuje HTML vlastním kódem, je založen na tvorbě komponent, nabízí funkci two-way binding a využívá virtuálního DOM. Hodí se na menší dynamické aplikace.

Výhodou Vue.js je velmi přehledná dokumentace, snadný syntax a malá paměťová náročnost. Nevýhodou je menší komunita a velká flexibilita, která může vést k různým přístupům. [41, 42]

2.5.6 Způsoby renderingu

V minulosti byla většina webových stránek statických bez žádných dynamických elementů. Celý HTML dokument pro každou stránku byl připraven na serveru k odeslání při dotazu klientské části. Dnes společnosti chtějí mít dynamický obsah a jeho správné zobrazení. Potřebují, aby se stránky načítaly rychle, měly dobré uživatelské rozhraní a dobrou optimalizaci pro vyhledávače (SEO). [45]

SEO (Search Engine Optimization) je proces zlepšování webové stránky pro zvýšení viditelnosti ve vyhledávačích. Lepší viditelnost zlepšuje povědomí o společnosti a službách, které nabízí. Více informací o tom, jak SEO funguje, je možné nalézt v [46].

V posledních letech se vyvíjejí i různé způsoby, jakými lze renderovat webové stránky. Mezi ně patří Client-side rendering, Server-side rendering a Static site generation.

2.5.6.1 Client-site rendering (CSR)

Tento přístup je nejvíce využíván na webu s velmi dynamickým obsahem. Načítání dat je řízeno JavaScriptem na straně klienta. Po požadavku klienta na odkaz na webovou stránku server pošle veškeré statické soubory (HTML a CSS) klientovi. Klient stáhne HTML a následně JavaScript, který vygeneruje obsah. Následně může uživatel plynule interagovat s webem. Poté již klient posílá na server požadavky pouze na obsah, nikoliv na statické soubory.

Kvůli stahování velkého množství souborů je počáteční načtení pomalejší, ale následné používání je rychlé. Protože je obsah generovaný JavaScriptem, může být přehlednut a nezahrnut při indexaci roboty vyhledávačů. Tato vlastnost negativně ovlivňuje SEO.

Dobrým případem použití je Single-page application (SPA). V SPA se každá stránka vykresluje v klientském prohlížeči. Server poskytuje pouze jeden HTML dokument. SPA vytvářejí frameworky jako je React, Angular nebo Vue.js. [45, 47]

2.5.6.2 Server-side rendering (SSR)

Tento přístup se používá na webové stránky s dynamickým personalizovaným obsahem. Je to skoro opak CSR. Stránky jsou již renderovány na serveru. Klient zadá požadavek, server zpracuje HTML, CSS a JavaScript a dodá klientovi již plně vyrenderovanou stránku. Na rozdíl od CSR se při každé další navštívené stránce proces renderování na serveru opakuje. To zpožďuje dobu načítání stránky.

Protože jsou stránky posílány na klienta již ve dokončeném stavu, mají lepší podporu SEO. Změna obsahu je oproti SSG vidět okamžitě, protože se stránky renderují přímo na serveru. [45, 47]

2.5.6.3 Static site generation (SSG)

Poslední přístup se nejvíce hodí na webové stránky s velkým množstvím obsahu, který se nemění moc často a nemusí být personalizovaný. Stránky se renderují už při sestavování pomocí generátorů statických stránek. Po požadavku se jen požadovaná stránka pošle klientovi. Toto může být velmi rychlé, protože se již nic nerenderuje po požadavku. Podobně jako SSR má lepší podporu SEO, protože stránky jsou také v dokončeném stavu, když jsou poskytovány klientovi.

Pokud však nastane změna obsahu, je nutné všechny stránky znovu sestavit, což může být u větších webů dlouhý proces. Proto není vidět změna obsahu hned. [45, 47]

2.6 Výběr řešení

Z analýzy vychází, že pro nový web je nejlepší využít headless CMS. Vzhledem k absenci frontendu otevírá velké možnosti pro vytvoření unikátní prezentační části a oproti vlastnímu řešení systému pro správu obsahu jsou již hotová řešení CMS bezpečná a udržovaná. Vybrané headless CMS bude jednoduše upravitelné, aby se zajistilo splnění požadavků F5 a F7, a bude zajišťovat distribuci dat pomocí API pro komunikaci s frontendem.

Prezentační část bude tvořena ve frameworku pro rychlejší vývoj a zabezpečení aplikace. Zároveň vybraný framework bude podporovat možnost SSG.

Jako možnost renderingu frontendu je nejlepší využít SSG, protože se obsah nebude měnit často. SSG umožňuje rychlejší načítání stránek, poskytuje lepší SEO a hostinky pro tento přístup jsou velmi levné.

Návrh

V této kapitole se zaměřím na návrh aplikace. Provedu a odůvodním výběr systému pro správu obsahu a technologií pro frontend. Dále se podívám na návrh rozložení prvků v prezentační části, který je proveden pomocí wireframe. Následně popíšu architekturu aplikace a důvod jejího použití.

3.1 Administrační část a backend

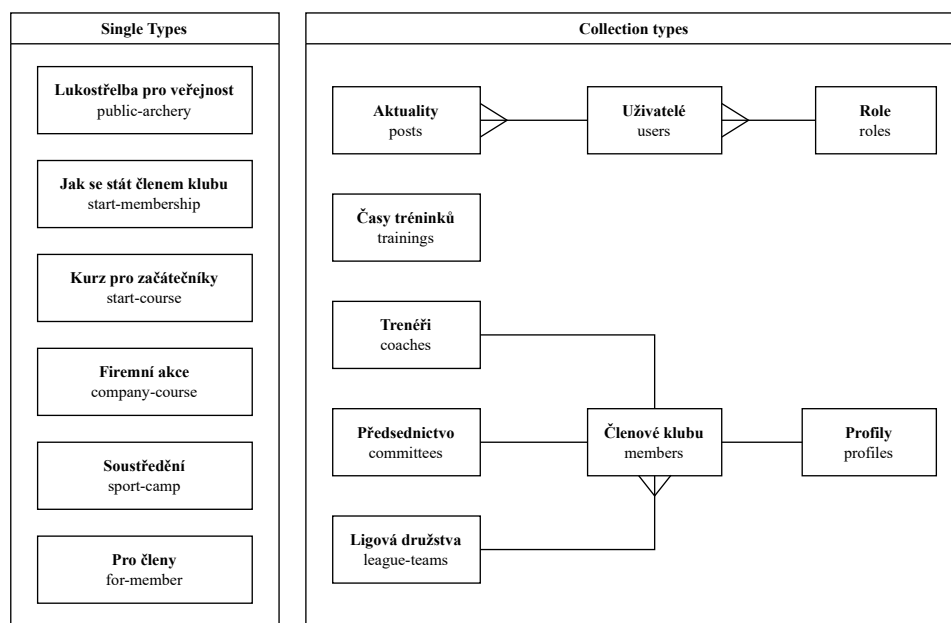
Hlavním kritériem výběru řešení backendu a administrační části byla cenová dostupnost a možnost velkého přizpůsobení. Toto splňují některé headless CMS a vlastní řešení systému pro správu systému. Ve vlastním řešení je nutné řešit bezpečnost, která je již v hotových řešeních zajištěna. Proto jsem nakonec ze všech možností vybrala Strapi.

3.1.1 Strapi

Strapi je open-source headless CMS, které je postavené na JavaScriptu a využívá MVC architekturu. Jako backend používá Node.js a pro uživatelské rozhraní administračního panelu používá React. Umožňuje úpravu administrační části a tvorby různých pluginů. Pro potřeby klienta je třeba implementovat načítání dat ze souboru a také automatické odesílání e-mailů na dané adresy. Díky open-source řešení mohu upravit nebo přidat části kódu, které budou zmíněné požadavky řešit. Strapi také umožňuje výběr databáze, do které se budou ukládat data. Jako výchozí databáze je nastavená SQLite a další možnosti výběru jsou: PostgreSQL, MySQL, MongoDB a MariaDB.

Struktura dat se definuje pomocí Content-Type Builder, kde se vytváří jednotlivé typy obsahu. Tyto typy mohou být jednorázové nebo v kolekci. Typy v kolekci (Collection Types) se vytvářejí, když je třeba opakovat stejný typ obsahu jako například aktuality nebo profily osob. Oproti tomu se jednorázové typy (Single Types) vytvářejí na data s jedinečnou strukturou jako je například obsah na domovské stránce. Dále se mohou vytvářet komponenty.

3. NÁVRH



Obrázek 3.1: Rozdělení typů obsahu na jednorázové a v kolekci

Komponenty jsou opakovatelně použitelné struktury, které se dají zahrnout ve více typech obsahu buď jako jediná položka, nebo jako seznam položek (například seznam sekcí nebo odkazy). Poslední stavební blok obsahu jsou dynamické zóny. Do dynamických zón se přidávají komponenty, ze kterých si následně správce obsahu může vybrat, a tím si přizpůsobit stránku za běhu.

Typy obsahu mohou propojovat pomocí relací stejně jako v relačních databázích. Každý typ obsahu má svá pole, která mají určitý typ jako například číslo, text nebo výčet. V obrázku 3.1 jsem rozdělila typy obsahu potřebné k tvorbě nového webu na jednorázové a v kolekci. Jsou zde zobrazené také použité relace.

Po vytvoření typu obsahu se automaticky vytvoří schéma, model a controller k danému typu. Tyto soubory se mohou libovolně upravit. Dále se musí nastavit oprávnění pro endpoint, zda ho může zavolat jen autorizovaný uživatel, nebo je to endpoint veřejný. Tato oprávnění se mohou nastavit na různé HTTP metody. Pokud oprávněný uživatel může vytvořit objekt do kolekce, zaškrtně se v aplikaci u daného endpointu políčko CREATE. Strapi distribuuje data pomocí REST API anebo GraphQL API.

Obsah se vytváří v nástroji Content Manager. Pro daný typ obsahu je zde příslušný odkaz v menu. Obsah se píše do příslušných polí. Po vytvoření obsahu pro typy v kolekci je možné obsah filtrovat, třídit a stránkovat a to jak v administračním panelu, tak i pomocí požadavků na API.

Dále má aplikace plugin na uživatelsky přívětivé nahrávání a ukládání ob-

rázků, videí, zvukových stop či dokumentů. Tento plugin se nazývá Media Library. Podporuje nahrávání obrázků pomocí URL či drag-and-drop funkce a umožňuje nahrát jeden či více obrázků najednou. Obrázky se dají ořezávat přímo v aplikaci a může se k nim přidat popis, který podporuje SEO. Pokud je potřeba změnit obrázek, který je na více stránkách, dá se jednoduše nahradit v galerii, a tím se automaticky změní i tam, kde je použitý. Plugin také optimalizuje velikost souborů bez ztráty dat a pro obrázky automaticky vytváří více formátů pro responzivní zobrazení. Media Library lze připojit na nějaký ze čtyř dostupných úložišť jako je například Cloudinary [48].

Strapi podporuje tvorbu vícejazyčného obsahu. Když je povolena tato funkce, obsah se může vytvářet ve stejném typu pro více jazyků. Může se také přizpůsobovat struktura pomocí komponent a dynamických zón. Tuto funkci v práci nevyužiji, protože web lukostřeleckého klubu je určený jen české veřejnosti.

Uživatelé administrační části mohou mít různé role. V neplacené verzi Strapi je na výběr ze tří rolí, u kterých lze nastavit jméno a práva, co může uživatel s touto rolí dělat. Práva mohou být nastavena ke každému typu obsahu, zda ho daný uživatel může vytvářet, číst, aktualizovat a mazat. Dále se pro roli může povolit správa pluginů a nastavení.

3.1.2 Databáze

Jak bylo zmíněno výše, Strapi umožňuje použití čtyř databázových systémů. Z nich jsem vybrala databázi PostgreSQL, protože je velmi rozšířená a podporuje ji velké množství hostingových služeb.

PostgreSQL je open-source objektově-relační databázový systém, který používá a rozšiřuje jazyk SQL. Jde o velmi stabilní systém, který je vyvíjen více než 30 let. Je používán jako primární úložiště dat pro velké množství aplikací. Pomáhá vývojářům vyvíjet aplikace a spravovat data neohledě na to, jak malá nebo velká je datová sada. [49]

3.2 Prezentační část

Z důvodu výběru backendu je nutné vytvořit prezentační část, která bude zpracovávat a prezentovat data, která přijme pomocí API. Pro potřeby této aplikace jsem vybrala framework Nuxt.js.

Důvodem výběru frameworku Nuxt.js je jeho jednoduchost, malá náročnost na paměť a mé předchozí zkušenosti s vývojem v něm. Navíc podporuje možnost SSG, kterou jsem vybrala, protože obsah se na stránkách nebude měnit příliš často a hostingy pro tento přístup jsou velmi levné.

3.2.1 Nuxt.js

Nuxt.js je open-source framework postavený nad Vue.js. Usnadňuje vývoj Vue aplikací a umožňuje jiné způsoby renderování než jen SPA, které nabízí Vue.js.

Pro základní použití frameworku se nemusí nic nastavovat, Nuxt.js má již mnoho věcí automaticky přednastavených. Oproti Vue.js se nemusí nastavovat soubor, kde se definují cesty na jednotlivé stránky. Routování je automatické pomocí systému souborů ve složce `pages`. Framework také poskytuje automatický import komponent a velké množství modulů, které usnadňují práci s různými aplikacemi či knihovnami, například s Axios, Bootstrap nebo Google Analytics. Dále obsahuje systém pro různá uspořádání stránek (layout). Pokud na webu existují podobné stránky, například příspěvky na blog, může se pro ně vytvořit samostatný layout, který se následně nastaví u příslušných stránek.

3.2.2 Vuex

V aplikaci bude třeba řešit správu stavu aplikace. Pro framework Vue.js, potažmo Nuxt.js, existuje knihovna Vuex, která umožňuje centralizovanou správu stavu pro všechny komponenty, a zajišťuje, že stav může být měněn pouze předvídatelným způsobem.

Vuex soubory se skládají z následujících funkcí a objektů: objekt *state*, který ukládá aktuální stav, funkce *mutations*, které jako jediné mění stav, funkce *actions*, které vyvolávají nějaký proces, například požadavek na server, a funkce *getters*, díky kterým lze získat stav aplikace a dále ho zpracovávat, aniž bychom měnili ten původní.

Ve Vuex existují moduly, díky nimž se může vytvářet více Vuex souborů, například pro každou funkčnost či komponentu jeden. Tyto moduly se mohou pojmenovat, a tím se vytváří jmenné prostory. V Nuxt.js se může jmenný prostor vytvořit pojmenováním složky a Vuex soubor se dá rozdělit do podsouborů `state.js`, `actions.js`, `mutations.js` a `getters.js`. [50]

3.2.3 Sass

Pro definování stylů jsem vybrala preprocesor Sass (Syntactically Awesome Style Sheets). Rozšiřuje syntaxi CSS a je kompatibilní se všemi jeho verzemi. Díky Sass lze jednoduše definovat proměnné, vnořovat hierarchii stylů, vytvářet moduly, které lze použít v jiných souborech, a mixiny, kterým lze předat proměnné pro větší flexibilitu, rozšiřovat definované styly a vypočítávat hodnoty pomocí matematických operátorů. [51]

3.2.4 Wireframe

Nedílnou součástí návrhu webové aplikace je wireframe. Jde o návrh kostry webu, jeho struktury, rozložení prvků, funkcionalit a chování. Jde o jakýsi

nákres, jak by web mohl vypadat, a pomocí něj si lze lépe představit navigaci a kompozici jednotlivých stránek. Wireframe reprezentuje první návrh aplikace, proto návrh grafické podoby jednotlivých sekcí je jen minimální. Často se tvoří jen ve stupních šedi.

Wireframe se tvoří poté, co se zanalyzují požadavky klienta, a předtím, než vývojáři začnou programovat uživatelské rozhraní. Po vytvoření se wireframe předvede klientovi, zda je návrh dle jeho vize, nebo zda by na návrhu něco předělal. Tato část, kdy si klient s vývojáři ujasní své vize, může ušetřit spoustu času a peněz. [52, 53]

Návrh jsem tvořila nejdříve na mobilní zařízení a pak na desktop. Tomuto přístupu se obecně ve web designu říká *mobile-first*. Mobilní zařízení mají totiž více restrikcí na design a strukturu než desktop.

Pro vytváření wireframů jsem využila nástroje Figma. Jde o webovou aplikaci pro vytváření grafického designu. Dají se zde kromě wireframů webových stránek vytvářet i designy mobilních aplikací, prototypy, příspěvky na sociální média a další. Výhodou je také možnost spolupráce více lidí na vytvářených projektech. [54]

Všechny wireframy, které jsem využila při návrhu a následné komunikaci s klientem, jsou na přiloženém médiu. Klient vytkl pár věcí, které jsem následně v návrzích předělala.

Každá stránka na webu má stejnou hlavičku a patičku. V hlavičce, která je zobrazena na návrhu stránky *Kontakt* v obrázku 3.6, se nachází logo klubu, pět položek navigace, které nejsou na menších zařízení zobrazené, a velké rozkliknutelné menu. Ve velkém menu se nachází pět sekcí. Návrh tohoto menu na desktopové zařízení je vidět v obrázku 3.2. Na mobilním zařízení jsou sekce zobrazeny ve sloupci.

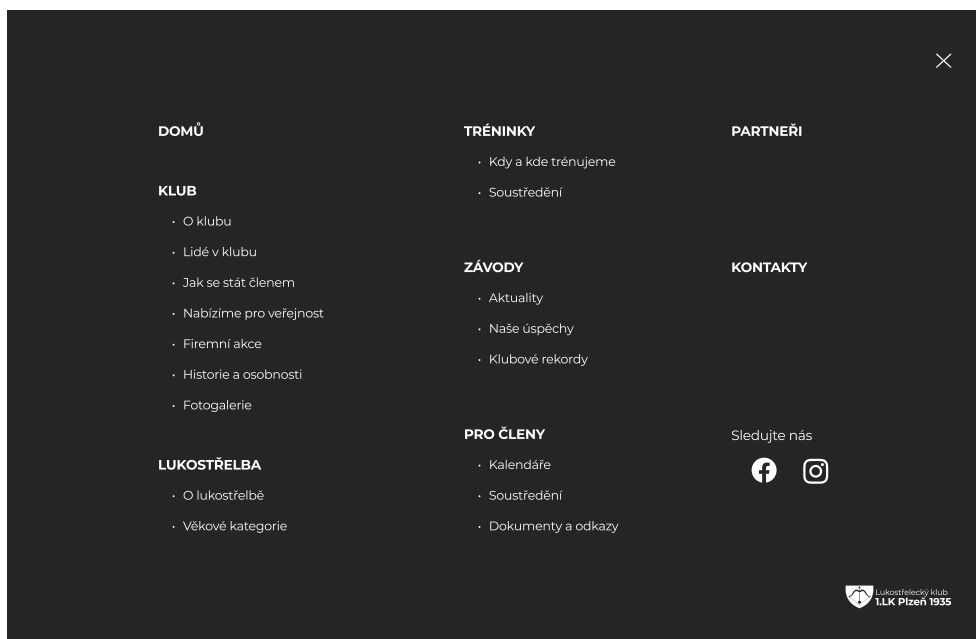
V patičce se nachází pět sekcí. První sekce jsou kontaktní údaje, tj. adresa, e-mail a telefonní číslo. Spolu s kontaktními údaji je v sekci ještě mapa na danou adresu klubu. Druhá sekce jsou odkazy na sociální sítě pod příslušnými ikonami. Třetí sekce jsou důležité odkazy na webové stránky, které se týkají lukostřelby nebo klubu. Čtvrtá sekce je logo klubu a pátá je sekce s logy a odkazy hlavních partnerů. Tyto sekce jsou na desktopovém zařízení ve třech sloupcích, na mobilních zařízeních v jednom. Patička je zobrazena v návrhu stránky *Kontakt* na obrázku 3.6.

Rozložení stránek se velmi změnilo. V tabulce 3.1 je vysvětleno mapování nových stránek a jejich sekcí na stránky a sekce v současném řešení webu 1. LK Plzeň 1935 [1]. Ve sloupci *Nová stránka* je název stránky na novém webu, ve sloupci *Sekce* je název sekce na příslušné stránce. Tyto stránky a sekce se mapují na staré stránky a sekce, které jsou napsané ve sloupci *Stará stránka – sekce*. V tomto sloupci je napsán vždy název stránky, za pomlčkou název sekce na této stránce a v závorkách poznámka. Pokud nová stránka přebírá z více stránek a sekcí, jsou tyto informace oddělené čárkou. Pokud je zde napsané slovo *není*, znamená to, že tato stránka či sekce na současném webu chybí.

3. NÁVRH

Nová stránka	Sekce	Stará stránka – sekce
Domovská stránka		není
Klub	O klubu	O klubu – O lukostřelbě (část textu)
	Lidé v klubu	Osazenstvo
Chci střílet z luku	FAQ	není, Info – Finanční stránka sportu
	Jak se stát členem	Veřejnost
	Nabízíme pro veřejnost	
	Firemní akce	
Historie	Historie	O klubu – Historie
	Osobnosti	není (O klubu – Osobnosti)
Fotogalerie		Foto, Video
Lukostřelba	O lukostřelbě	O klubu – O lukostřelbě (část textu)
	Věkové kategorie	není
Tréninky	Kdy a kde trénujeme	Info
	Soustředění	
Závody	Aktuality	Domovská stránka
	Naše úspěchy	O klubu – Úspěšnost
	Klubové rekordy	není
Pro členy	Kalendáře	Postranní panel
	Soustředění	není
	Dokumenty a odkazy	Ke stažení, Partneři – odkazy
Partneři		Partneři
Kontakty		Kontakty

Tabulka 3.1: Mapování nového rozložení stránek na stávající rozložení na webu 1. LK Plzeň



Obrázek 3.2: Návrh velkého menu

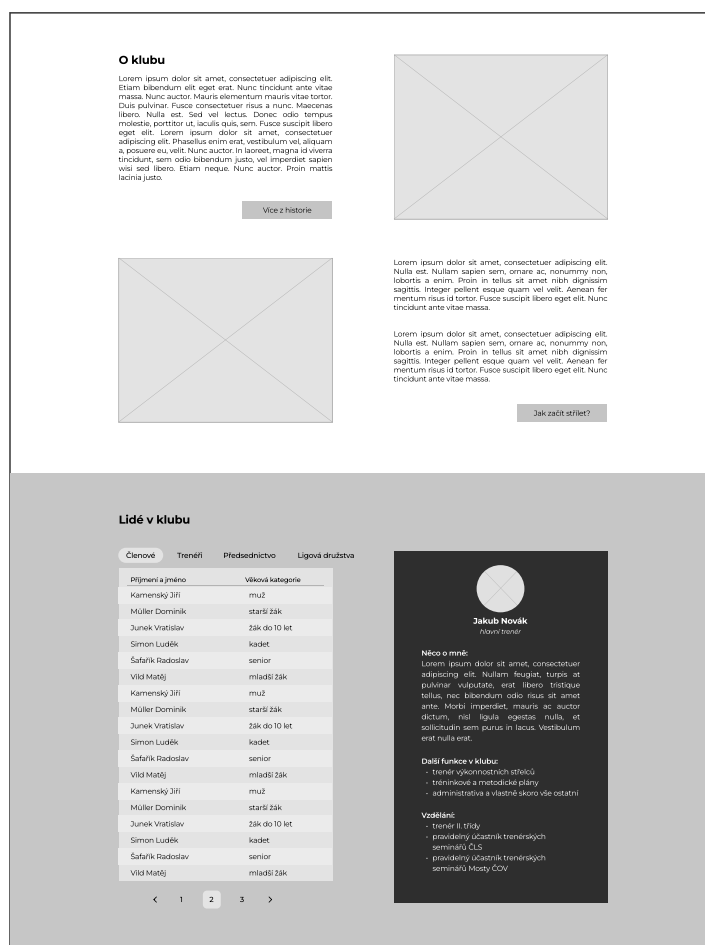
V následujících sekcích popíši pár vybraných wireframů, které jsou něčím zajímavé či odlišné od současného webu. Ukázky wireframů jsou rozloženy na desktop.

3.2.4.1 Domovská stránka

Domovská stránka je rozdělena do šesti sekcí. První sekcí je úvod stránky, který se skládá z obrázku přes celou stránku a motivačního textu s odkazem na sekci pod úvodem. Následuje sekce s krátkým popisem klubu a odkazem na stránku *Klub*, který je zobrazen pomocí tlačítka. Podobná je sekce o lukostřelbě, která následuje. Dále je tam sekce s profily významných členů, kteří se podílí na chodu klubu. Tyto profily obsahují informace o osobě a budou se posouvat automaticky nebo kliknutím na šipky. Také je využiji na stránce *Klub* v sekci *Lidé v klubu*. Předposlední sekce obsahuje tři poslední aktuality. Karta jedné aktuality obsahuje titulní fotografii, začátek textu a odkaz na stránku s aktualitou. Tyto karty využiji také na stránce *Závody* v sekci *Aktuality*. V poslední sekci se nachází loga partnerů, která obsahují odkazy na jejich webové stránky.

Návrh této stránky se klientovi líbil, jen jsem zde prohodila mezi sebou sekce *lukostřelba* a *klub*. Návrh je již ve finální verzi, a protože je stránka dlouhá, tak je návrh pouze na přiloženém médiu.

3. NÁVRH

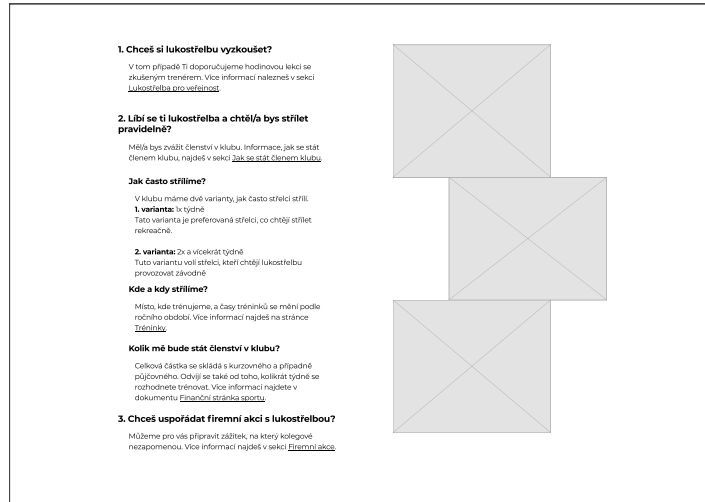


Obrázek 3.3: Návrh stránky *Klub*: sekce o klubu a lidé v klubu

3.2.4.2 Klub

V návrhu (obrázek 3.3) tato stránka obsahuje informace o klubu a odkazy na stránku s historií klubu a na stránku s informacemi, jak se stát členem klubu. Následující sekce je rozdělena na dva sloupce. První sloupec se seznamy lidí v klubu, kteří jsou rozděleni na čtyři kategorie: členové klubu, trenéři, předsednictvo a ligová družstva. Seznam se členy klubu bude stránkovaný. Tyto seznamy jsou i na současném webu, ale jsou rozděleny na jiné kategorie.

Druhý sloupec využije profily lidí z domovské stránky. Zde se také budou posouvat automaticky nebo pomocí šipek. Profily budou obsahovat odkaz na stránku s větším profilem. Stránka s profilem bude u ligových střelců obsahovat také odkaz na osobní profil na stránce Rcherz [6]. Profily s fotografiemi osob a odkazy na Rcherz jsem se inspirovala stránkou *Závodníci* na webu lukostřeleckého oddílu TJ Opava popsaného v sekci 2.3.1.



Obrázek 3.4: Návrh stránky *Chceš si zastřílet z luku*: sekce otázky a odpovědi



Obrázek 3.5: Návrh stránky *Chceš si zastřílet z luku*: sekce jak se stát členem klubu

3.2.4.3 Chceš si zastřílet z luku?

Tento návrh stránky obsahuje čtyři sekce. První sekcí jsou otázky ohledně možností střelby z luku a odpovědi na ně. V této části jsem se inspirovala ve dvou částech popsaných webů v rešerši webů s podobnou tematikou, přesněji u sekce FAQ na stránce Redhill Archers popsané v sekci 2.3.3 a u aktuality s otázkami a odpověďmi na stránce Lukostřelba Brno popsané v sekci 2.3.2. V odpovědích jsou také odkazy na sekce níže. Návrh na tuto část je zobrazena v obrázku 3.4.

Druhá sekce obsahuje informace o lekcích pro veřejnost a formulář, který slouží pro zájemce o tuto lekci. Třetí sekce je velmi podobná druhé, také obsahuje formulář. Dále obsahuje informace o tom, jak se stát členem, a o kurzu pro začátečníky spolu s termíny, kdy jsou kurzy vypsané. Tato sekce je zobrazena v obrázku 3.5. Poslední lekce obsahuje informace o firemních akcích spolu s kontaktem na odpovědnou osobu.

3.2.4.4 Kontakt

V návrhu této stránky, který je zobrazen v obrázku 3.6, jsou tři sekce. První sekce je inspirovaná sekcí na stránce *Kontakt* webové stránky Lukostřelby Brno popsané v sekci 2.3.2. Jedná se o krátký popis místa, kde se nachází střelnice, adresa, dodatečné informace a je zde přiložená mapa. Druhá sekce je kontaktní formulář, který na současném webu chybí a je obsažen na všech popsaných webech v sekci 2.3. Kontaktní formulář usnadňuje uživateli cestu, jakou se může na něco zeptat, a proto jsem se rozhodla ho na nový web implementovat. Poslední sekcí jsou obecné kontakty jako je hlavní e-mail klubu, telefonní číslo na šéftrenéra, korespondenční adresa, bankovní spojení, IČ a spisová značka.

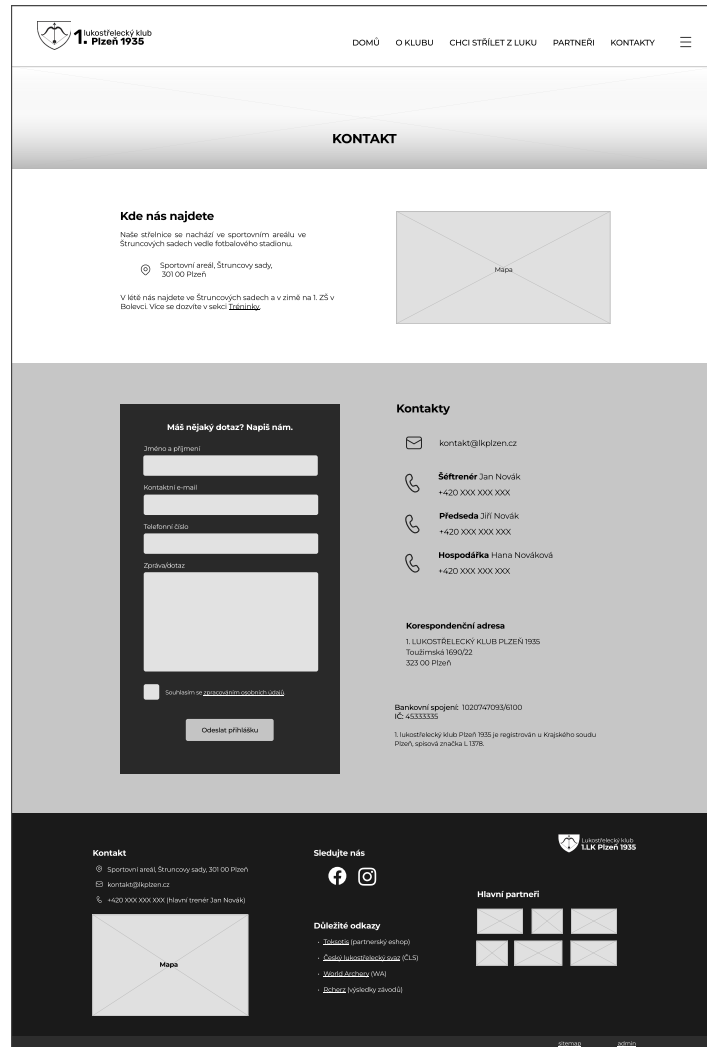
3.2.5 Design

Cílem nového frontendu je udělat modernější webové stránky. Dobrým příkladem jsou stránky klubu Redhill Archers, které jsou popsány v sekci 2.3.3. Existuje web Webdesign Inspiration [55] s příklady webového designu, který je prezentován odkazy na webové stránky se zajímavým a něčím unikátním designem. Právě nápady z tohoto webu se ve směru designu budu velmi inspirovat.

3.3 Architektura aplikace

Vzhledem k vybranému řešení systému pro správu obsahu bude zvolena architektura MVC. Strapi je totiž postavené na této architektuře a bylo by náročné až skoro nemožné tento návrh změnit.

3.3. Architektura aplikace



Obrázek 3.6: Návrh stránky *Kontakt*

3.3.1 Architektura MVC

MVC je architektonický vzor, který odděluje aplikaci do tří hlavních komponent: Model, View a Controller. Každá z těchto komponent je vytvořena tak, aby zpracovávala specifické aspekty vývoje aplikace. [56]

Model odpovídá veškeré logice související s daty. Jde o nejnižší úroveň aplikace, která je propojená s databází, a všechny operace s daty jdou právě přes tuto komponentu. V této aplikaci je ve Strapi pro každý typ obsahu automaticky vygenerována komponenta Model, která komunikuje s vybranou databází.

View se používá pro logiku uživatelského rozhraní aplikace. Má za úkol v čitelné formě prezentovat data. Může obsahovat komponenty uživatelského rozhraní, jako jsou například textová pole nebo rozbalovací nabídky, se kterými pracuje koncový uživatel. V této aplikaci se jedná o prezentační a administrační část.

Controller funguje jako rozhraní mezi komponentami Model a View. Zpracovává veškerou business logiku a příchozí požadavky, manipuluje s daty pomocí komponenty Model a komunikuje s komponentou View při renderingu konečného výstupu. V této aplikaci je ve Strapi pro každý typ obsahu vytvořena komponenta Controller, kterou lze modifikovat.

3.3.2 Komunikace mezi prezentační částí a backendem

Protože je prezentační část staticky generovaná, je potřeba při změně obsahu na backendu předat informaci frontendu, že se musí znovu sestavit, aby se změny projevily. Pro tuto situaci se využívá webhook. Jde o cestu, jak jedna aplikace poskytuje informace v reálném čase druhé aplikaci. V tomto případě je to informace o změně obsahu. [57]

Strapi nabízí možnost dvou různých API pro distribuci dat: REST API a GraphQL API. Pro tuto aplikaci jsem vybrala REST API.

3.3.2.1 REST

REST (Representational State Transfer) [58] je softwarová architektura pro distribuované hypermediální systémy. Má vlastní principy a omezení, které musí být splněny, aby se API mohlo říkat RESTful. Jsou to následující principy:

Obecné rozhraní

Všechny API požadavky na jeden zdroj musí vypadat stejně bez ohledu na to, odkud požadavek přišel. REST API musí zajistit, aby stejné údaje patřily pouze jednomu URI (jednotný identifikátor zdroje). Zdroje by neměly být příliš velké, ale měly by obsahovat všechny informace, které by mohl klient potřebovat.

Klient-server

Při návrhu REST API musí být klientská a serverová aplikace na sobě zcela nezávislá. Jediná informace, kterou by měl klient znát, je URI požadovaného zdroje. Server by neměl modifikovat klientskou aplikaci jinak než předáním požadovaných dat prostřednictvím protokolu HTTP.

Bezstavovost

REST API je bezstavové, to znamená, že každý požadavek musí obsahovat všechny informace potřebné pro jeho zpracování. Neměly by se ukládat na straně serveru žádné informace související s požadavky klienta.

Možnost ukládání do mezipaměti

Pokud je to možné, měly by být zdroje ukládány v mezipaměti na straně klienta nebo serveru. Odpovědi serveru by měly také obsahovat informaci, zda je povoleno zdroj do mezipaměti ukládat. Cílem je zlepšit výkon na straně klienta a zvýšit škálovatelnost na straně serveru.

Vrstvená architektura

V REST API procházejí požadavky a odpovědi různými vrstvami. Rozhraní musí být navrženo tak, aby klientská ani serverová aplikace nepoznala, zda komunikují s koncovou aplikací, nebo s prostředníkem.

Kód na vyžádání (code on demand)

Jedná se o volitelný princip. REST API obvykle odesílá statické zdroje, ale v některých případech mohou odpovědi také obsahovat spustitelný kód. [58]

Implementace

Tato kapitola popisuje postup implementace administrační a prezentační části. V administrační části se zaměřím na nastavení a konfiguraci, následnou tvorbu typů obsahu a další rozšiřující funkcionality, které jsem v projektu implementovala.

V prezentační části popíši nastavení a strukturu projektu, dále se zaměřím na rozložení stránek aplikace a importované moduly. Poté vysvětlím účel jednotlivých komponent a stránek, jaké další funkcionality využívají a jak získávají data z backendu.

4.1 Administrační část a backend

Jak už bylo popsáno v návrhu, pro administrační část jsem zvolila headless CMS Strapi. V této sekci popíšu instalaci a počáteční nastavení aplikace, typy obsahu, které jsem v rámci aplikaci vytvořila, funkcionality, které jsem přidala a přizpůsobení administračního panelu.

4.1.1 Počáteční nastavení

Instalace Strapi probíhá přes npm, což je správce balíčků pro Node.js (více informací v [59]). Při instalaci systému se může nastavit konkrétní databáze, která se bude v projektu používat. Pro tuto aplikaci jsem zvolila a nastavila databázi PostgreSQL.

4.1.2 Typy obsahu

Pro určení typů obsahu jsem použila nástroj Content Type Builder, který Strapi obsahuje. Pro každý vytvořený typ se v projektu ve složce `/src/api` vytvoří složka s jeho jménem, která obsahuje: schéma v JSON a soubory `controller`, `service` a `route` soubor, které obsahují jen volání funkce, jenž vytvoří základní funkcionality daného souboru (například u `controlleru` to jsou funkce

find, create, update, delete a soubor route definuje všechny všechny metody, které lze na endpoint zavolat). Typy jsem tvořila dle návrhu a následně popíšu jejich funkci a atributy.

Každý typ obsahu i komponenta mají atribut `id`. Dále má každý typ obsahu atributy `createdBy` a `updatedBy`, které uchovávají informaci o tom, jaký uživatel vytvořil a naposledy aktualizoval obsah v daném typu obsahu, a jsou relací propojeny s entitou `User`, což je automaticky vytvořená entita pro ukládání informací o uživateli, kteří mají přístup do systému. Poslední automaticky přidávané atributy jsou `createdAt` a `updatedAt`, do kterých je uloženo datum a čas, kdy byl obsah v typu obsahu vytvořen a naposledy aktualizován.

V jednorázových typech obsahu také využívám komponenty. Vytvořila jsem pro potřeby aplikace následující:

Course Information se používá pro doplňující informace o kurzu. Jedná se o dodatečné informace, termín, kdy kurz probíhá a cenu za absolvování.

Course Term se používá pro uchování informací pro jeden termín kurzu pro začátečníky. Tuto komponentu využívám pro konkrétní kurz, který má vždy čtyři lekce, a tak se zde pomocí atributů `firstTime`, `secondTime`, `thirdTime` a `fourthTime` textově zaznamenává datum a čas jednotlivých lekcí.

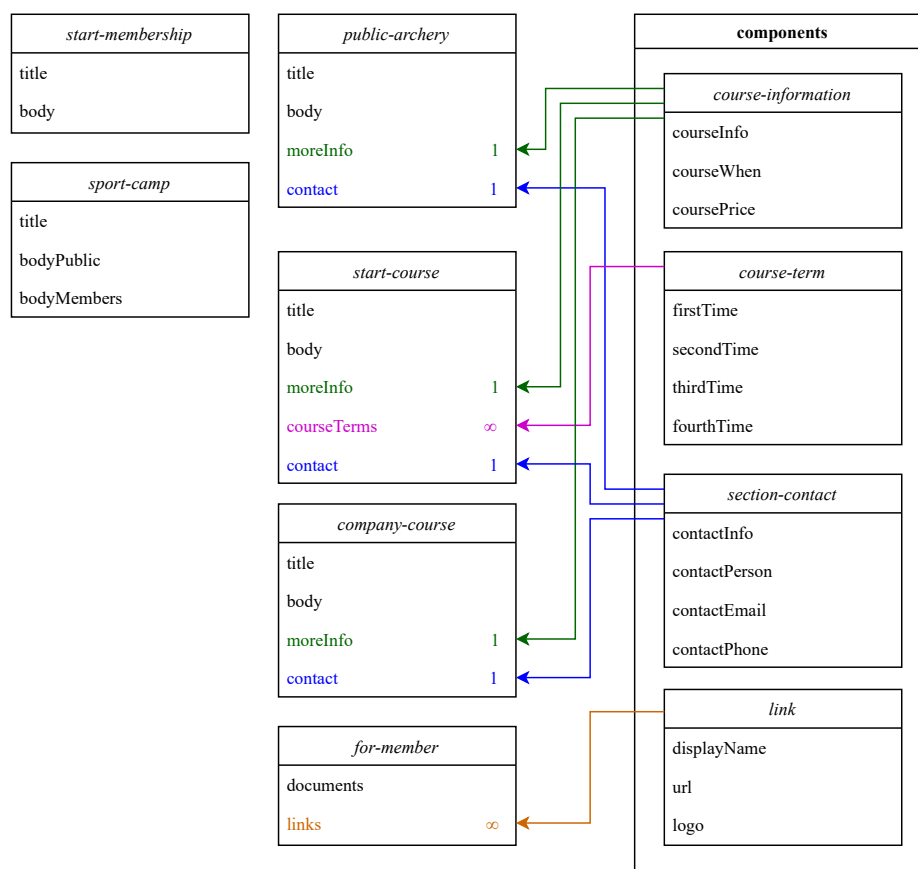
Section Contact přidává k typu obsahu, ve kterém je použit, kontaktní údaje osoby odpovědné za službu popsanou v dané sekci. Jedná se o informace ke kontaktu, jméno a příjmení odpovědné osoby, její e-mailovou adresu a telefonní číslo.

Link se používá pro seznamy odkazů na jiné webové stránky. Komponenta obsahuje titulek, který se bude zobrazovat na webu, URL adresu na daný odkaz a popřípadně logo stránky, na kterou odkaz vede.

4.1.2.1 Single Types

Jednorázové typy obsahu (Single Types) se vytvářejí, pokud daná stránka má jedinečnou strukturu, která se jinde v aplikaci nevyskytuje. Pro potřeby aplikace jsem vytvořila šest jednorázových typů obsahu, u kterých v následujícím seznamu popíšu jejich funkci na webových stránkách a uvedu atributy, z jakých se skládají. Všechny jednorázové typy spolu s atributy a komponentami jsou zobrazeny v diagramu 4.1.

Jak se stát členem klubu (start-membership) popisuje podmínky, za jakých se lze stát členem klubu. Skládá se jen ze dvou informací, a to z názvu a textu sekce.



Obrázek 4.1: Jednorázové typy obsahu, jejich atributy a komponenty

Soustředění (*sport-camp*) se skládá s názvu sekce a dalších dvou textů. První z nich jsou informace o soustředění pro veřejnost a druhý text jsou informace o soustředění pro členy.

Lukostřelba pro veřejnost (*public-archery*) informuje o pořádání lekcí pro veřejnost. Tento typ obsahu se skládá z názvu a textu sekce, z komponenty *Course Information*, která zobrazuje více informací o kurzu a je v typu obsahu obsažena pouze jednou (číslo 1 v diagramu 4.1), a z komponenty *Section Contact*, která slouží pro přidání kontaktní osoby k dané sekci a také je obsažena v typu obsahu pouze jednou.

Kurz pro začátečníky (*start-course*) popisuje informace o kurzu pro začátečníky, jeho vypsaných termínech a o kontaktní osobě. Obsahuje název a text sekce a komponenty *Course Information* pro zobrazení více informací o kurzu a *Course Term*, která slouží na výpis termínů pro tento

kurz. Termínů může být více (v diagramu 4.1 symbol ∞). Dále obsahuje komponentu *Section Contact* pro přidání kontaktní osoby na dotazy ohledně dané sekce.

Firemní akce (company-course) informuje o možnosti uspořádání firemní akce s lukostřelbou. Tento typ obsahuje stejné atributy a komponenty jako jednorázový typ obsahu *Lukostřelba pro veřejnost*, jen z komponent využívá jiné atributy.

Pro členy (for-member) obsahuje důležité dokumenty a odkazy, které by měli členové klubu znát, například dokument s provozním řádem sportoviště. Skládá se ze seznamu důležitých dokumentů a z komponenty *Link*, která obsahuje odkazy na důležité stránky. Odkazů může být více a dokument může být v různých formátech.

4.1.2.2 Collection Types

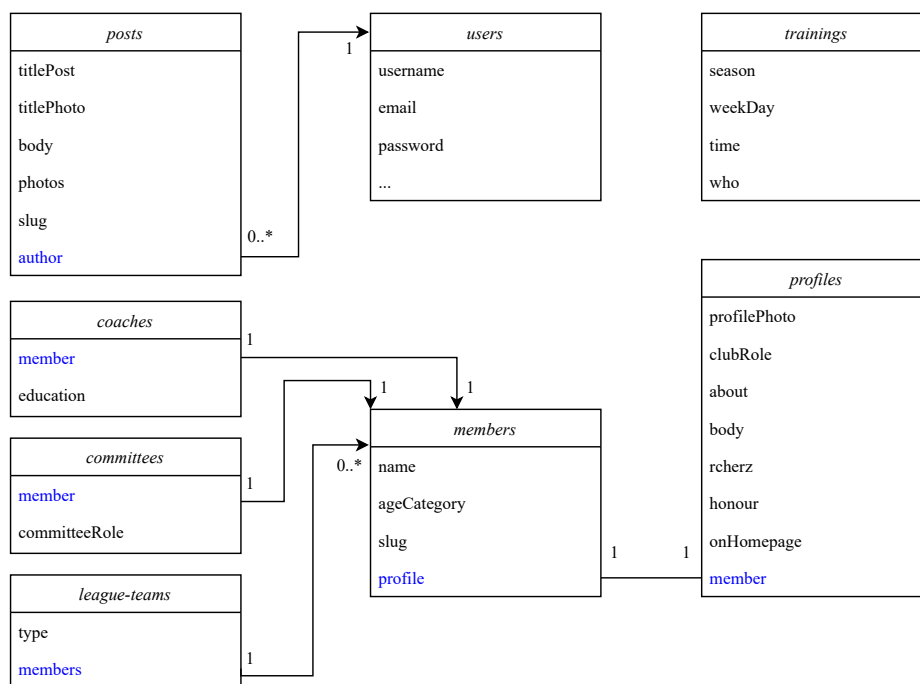
Typy obsahu v kolekci (Collection Types) se vytvářejí pro opakující se obsah, například aktuality nebo profily osob. Pro potřeby aplikace jsem vytvořila sedm typů obsahu v kolekci, osmý typ *Uživatelé* (users) je automaticky generován systémem Strapi pro uchovávání dat ohledně uživatelů, kteří mají přístup do systému. Všechny typy obsahu v kolekci jsou zobrazeny v diagramu 4.2. Jsou zde i zobrazeny relace mezi jednotlivými typy. V následujícím seznamu píšou funkci každého typu a jeho atributy.

Aktuality (posts) slouží pro vytváření aktualit ohledně závodů. Každá aktualita obsahuje tyto informace: titulek, titulní fotografii, text, doplňující fotografie, slug, což je automaticky vygenerovaný řetězec pro jednoznačnou identifikaci aktuality (více o jeho nastavení v sekci 4.1.3.1), a autora, který je vždy vybrán ze seznamu uživatelů. Aktualita může mít jen jednoho autora.

Tréninky (trainings) slouží pro výpis jednotlivých tréninků v týdnu. Tréninky jsou rozděleny na zimní a letní sezónu a obsahují tyto informace: den v týdnu, čas od kolika do kolika hodin trénink probíhá a informaci o tom, jaké věkové skupině je daný trénink určený.

Členové (members) je typ obsahu pro záznam všech členů klubu. Každý člen má jméno, věkovou kategorii a slug z vlastního jména. Dále obsahuje profil, který je přiřazen po vytvoření profilu dané osoby. Obsah se do tohoto typu obsahu bude moci přidávat pomocí importu souboru CSV.

Profily (profiles) slouží jako osobní profil přiřazeného člena. Obsahuje profilovou fotografii, krátký popis a delší popis osoby, odkaz na profil na webu Rcherz, a informaci, zda je člen významná osoba a zda se má jeho profil zobrazovat na domovské stránce mezi vybranými profily. Dále také



Obrázek 4.2: Typy obsahu v kolekci, jejich atributy a relace mezi nimi

obsahuje informaci o tom, jakému členovi profil patří. Ten se vybírá ze seznamu členů klubu a profil může patřit pouze jednomu členovi.

Trenéři (*coaches*) je seznam všech trenérů v klubu. Každý záznam obsahuje osoba, která je vybrána ze seznamu členů klubu, a jeho vzdělání na trenérské pozici.

Předsednictvo (*committees*) je seznam osob, které jsou v předsednictvu klubu. Každý záznam také obsahuje člena ze seznamu a jeho roli v předsednictvu.

Ligové týmy (*league-teams*) se rozdělují na tři týmy, které mají nějaký počet ligových střelců. Obsahuje vždy název ligy a seznam střelců, kteří jsou vybráni z řad členů klubu. Jeden záznam v tomto typu tak může mít více vybraných členů.

4.1.3 Rozšíření funkcionality

Abych splnila požadavky klienta, musela jsem základní aplikaci rozšířit o další funkce a pluginy. V následujících sekcích popíšu každou přidanou funkcionality.

4.1.3.1 Slug

Pro výstižnější a bezpečnější URL jsem se rozhodla využít slug systém pro aktuality a profily, které budou mít každý vlastní stránku. Slug pro profil tvořím v typu obsahu *Člen*.

K použití slug systému ve Strapi je možnost využít plugin Slugify. Plugin umožňuje automatické vytvoření slug řetězce z jakéhokoliv pole (či více polí) v typu obsahu. Dále lze po jeho nastavení najít jeden záznam podle slugu.

Pluginy se nastavují v souboru `/config/plugins.js`. Pro plugin Slugify se zde musí nastavit typ obsahu, pole, do kterého se bude automaticky vytvářet slug, a pole, ze kterého je slug vytvářen. Nastavila jsem zde tedy dva typy obsahu: aktualitu (atribut `slug` a jako referenční atribut `titlePost`) a člena (atribut `slug` a jako referenční atribut `name`).

4.1.3.2 Odesílání e-mailů

Pro splnění požadavku F7 je potřeba implementovat automatické odesílání e-mailů poté, co na server přijde požadavek s daty z formuláře. Ke správnému fungování se musí ve složce `/src/api` vytvořit nová složka s názvem typu obsahu, v mém případě je to `message`. Uvnitř této složky jsem vytvořila dvě podsložky se soubory `controller` a `route`. V `controlleru` definuji tři funkce (`public`, `start` a `contact`), každá z nich má na starosti jiný formulář. Funkce obsahuje validaci dat, definici e-mailových šablon a funkci na odeslání e-mailů. Soubor `route` definuje všechny endpointy, HTTP metody k nim a funkce, které mají na starost zpracování dat přijatých požadavkem na daný endpoint.

Aby fungovalo posílání e-mailů, musela jsem nastavit plugin `Email`. Nastavuje se zde poskytovatel a jeho konfigurace. Jako poskytovatele jsem zvolila `Nodemailer`, který využívá `SMTP` a není závislý na použité službě. Jako službu jsem využila `Sendinblue` [60], který poskytuje `SMTP` server a 300 e-mailů denně zdarma s možností příplatku pro neomezené množství. V konfiguraci poskytovatele jsem nastavila adresu `SMTP` serveru, jeho port a autorizační údaje. Dále se zde může nastavit výchozí adresu, ze které se e-mail posílá, a také adresu, která se uživateli objeví, pokud bude chtít odpovědět.

4.1.3.3 Import dat ze souboru

Import dat ze souboru, který je obsažen v požadavku F5, se mi nepodařilo implementovat. V předchozí verzi systému Strapi existoval plugin právě na import a export dat, ale ten v nejnovější verzi Strapi 4 nefunguje.

Existuje možnost samostatného řešení, které je ovšem velmi netriviální. Lze vytvořit vlastní plugin, který umožňuje přidání UI do samostatné karty ve Strapi. Pro nahrání souboru je možnost využít pluginu `Upload`, který však soubory nahrává do *Media Library*, což v tomto případě není žádoucí. V následujícím kroku by se musel zpracovat formát souboru a jednotlivé záznamy

nahrát do příslušného typu obsahu. Pro každý záznam by se zavolala metoda `create` na danou entitu typu obsahu.

Vzhledem k časové náročnosti tohoto řešení je část s importem dat ponechána na pokračování projektu.

4.1.4 Přizpůsobení administrátorského panelu

Ve Strapi lze administrační panel ve velké části přizpůsobit. Tuto možnost jsem využila pro překlad názvů typu obsahu, komponent a jejich atributů.

Veškeré modifikace se nastavují v souboru `/src/admin/app.js`. V tomto souboru jsem nastavila jazyk na český a vytvořila objekt, kde vždy definuji daný název a jeho překlad. Dále jsem zde nastavila vlastní logo a favicon, což je ikona, která se zobrazuje v kartě prohlížeče.

4.2 Prezentační část

Prezentační část je tvořena ve frameworku Nuxt.js. Tento framework se dá jednoduše nainstalovat pomocí npm. Aplikace je z velké části tvořena Vue soubory, které obsahují HTML, JavaScript i CSS. Každý soubor se většinou stará jen o svou část aplikace.

4.2.1 Konfigurace a souborový systém

Protože používám SSG, je nutnost nastavit tuto možnost v konfiguračním souboru `nuxt.config.js`. Zde se kromě způsobu renderování nastavují i defaultní hlavičky HTML souborů, importují globální CSS styly, konfiguruji importované moduly a mnoho dalších nastavení, které nevyužiji.

Mimo konfiguračního souboru je projekt rozdělen do několik složek. Každá složka má jinou funkci, kterou v následujícím seznamu krátce popíšu.

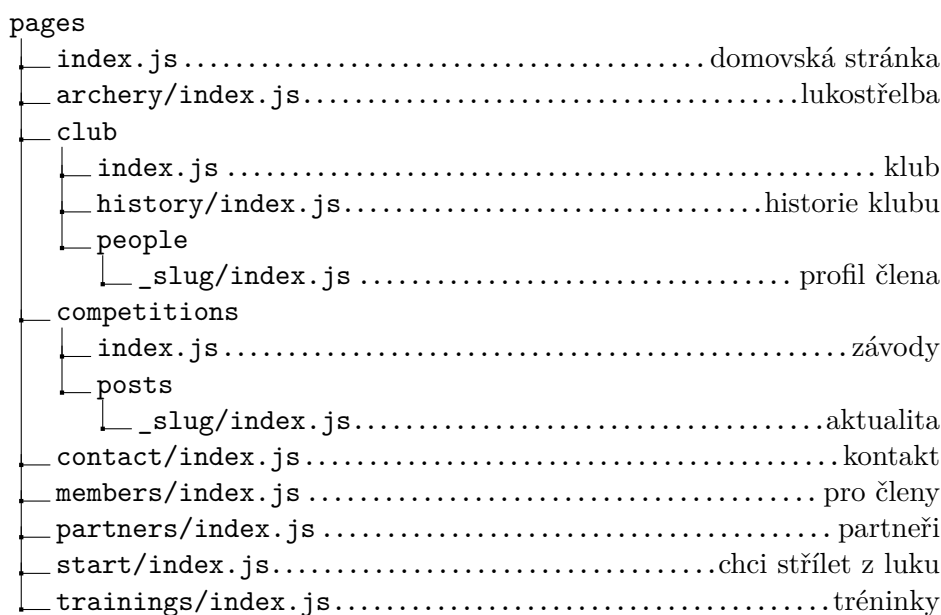
Assets obsahuje všechny doplňující soubory pro vzhled frontendu. Jsou zde soubory se styly, ikony, loga a fotografie.

Components obsahuje všechny Vue komponenty, které se používají v různých částech webu, mohou být použitelné vícekrát a importované v jakékoliv části projektu.

Layouts definuje různá rozložení webu. Defaultní rozložení je přednastavené pro každou stránku, pokud se v její konfiguraci neurčí jinak. Můžou se zde vytvářet vlastní rozložení pro nějaký typ stránek.

Pages je složka, ve které se definuje struktura aplikace. Podle této struktury je automaticky vygenerované routování mezi stránkami.

Static obsahuje soubory, které se nemění a mohou být volně přístupné. Lze je získat pod kořenovým URL a názvem souboru.

Obrázek 4.3: Systém souborů ve složce `pages`

Store obsahuje Vuex soubory. Jak bylo popsáno v sekci 3.2.2 Vuex se ve Vue aplikacích používá pro správu stavu aplikace. Využívá se také pro získávání dat ze serveru a jejich následné zpracování.

4.2.2 Vývoj

Na začátku vývoje ve frameworku Nuxt.js se definuje struktura aplikace, tj. které stránky budou na jaké URL a jaké bude mít podstránky. Ve složce `pages` jsem proto definovala strukturu, která je zobrazena v 4.3.

Dále jsem definovala pomocí CSS a Sass souborů obecné styly, jako je například font nebo barvy webu. Tyto soubory jsem přidala do konfiguračního souboru `nuxt.config.js` a nastavila modul, který umožňuje import Sass souborů a přímé psaní Sass syntaxe do Vue souborů.

Protože některá data z backendu jsou ve formě značkovacího jazyka Markdown, je nutné tato data převést do HTML. Proto jsem využila modul `Markdownit` [61], který pomocí syntaxe `$md.render()` převede Markdown do HTML prvků.

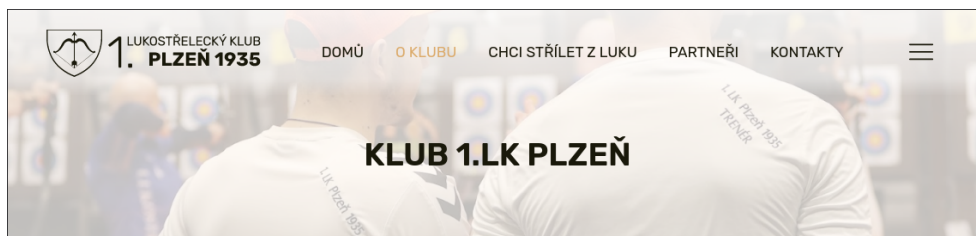
V následujících sekcích se podívám na tvorbu jednotlivých stránek a jejich komponent. Popíšu, jaké komponenty stránka využívá, zda a jakým způsobem používá Vuex.

4.2.2.1 Základní komponenty

Pro účely aplikace jsem vytvořila sedm hlavních komponent. Komponenty pro hlavičku, navigaci, velké menu a patičku jsem využila ve výchozím layoutu, a tak se objevují na každé stránce. Zbylé tři komponenty se objevují v různých částech napříč aplikací.

Hlavička

Pro hlavičku jsem vytvořila komponentu *TheHeader*, která obsahuje logo s názvem klubu a komponentu navigace (obrázek 4.4). Hlavička se nachází v horním okraji každé stránky a po sescrollování zůstává viditelná. Tato vlastnost je způsobena JavaScriptem, který sleduje pozici stránky od vrcholu, a pokud je tato pozice větší než 50 obrazových bodů, přidá se hlavičce třída, která je pomocí CSS zafixována k vrcholu viditelného okna.



Obrázek 4.4: Komponenta *TheHeader* a *Head*

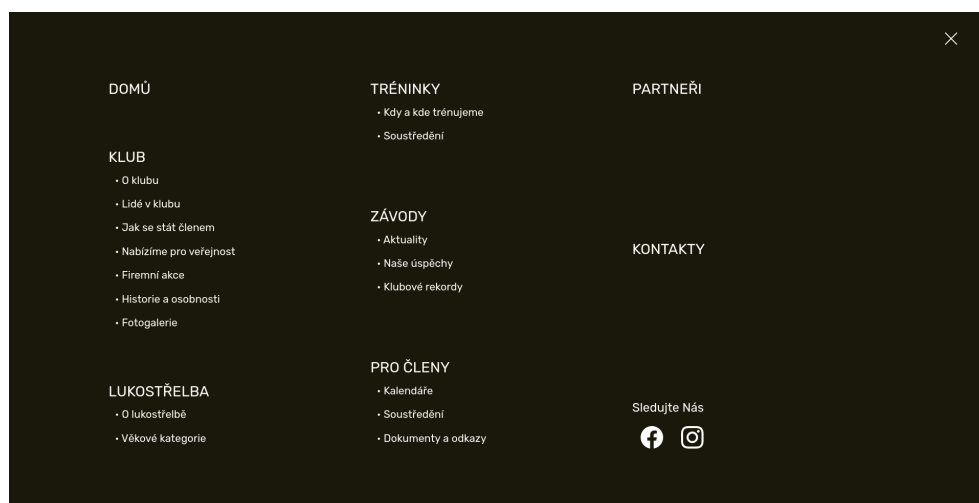
Navigace

Komponenta *Navigation* obsahuje pět hlavních položek menu a tlačítko na zobrazení velkého menu. Odkazy na další stránky jsou přidány pomocí elementu *nuxt-link*, který umožňuje navigaci mezi stránkami aplikace. Dále je zde přidána komponenta *HamburgerMenu*, která se zobrazí po kliknutí na tlačítko s ikonou menu. Tato komponenta je uvnitř *transition* bloku, který přidává animaci při zobrazení a skrytí komponenty. Pět hlavních položek menu je v menších zobrazení skryto.

Velké menu

Tento prvek je přidán pomocí komponenty *HamburgerMenu* a zobrazuje se jako celostránkové menu obsahující všechny odkazy na stránky a jejich sekce (obrázek 4.5). Menu je rozděleno do skupin podle významu a obsahuje odkazy, které jsou též přidány díky elementu *nuxt-link*. Položka *Fotogalerie* nemá odkaz, protože fotografie se budou skladovat v externím úložišti, které není ještě zřízeno. Pro skrytí menu je zde tlačítko s ikonou kříže, který zavolá funkci z nadřazené komponenty.

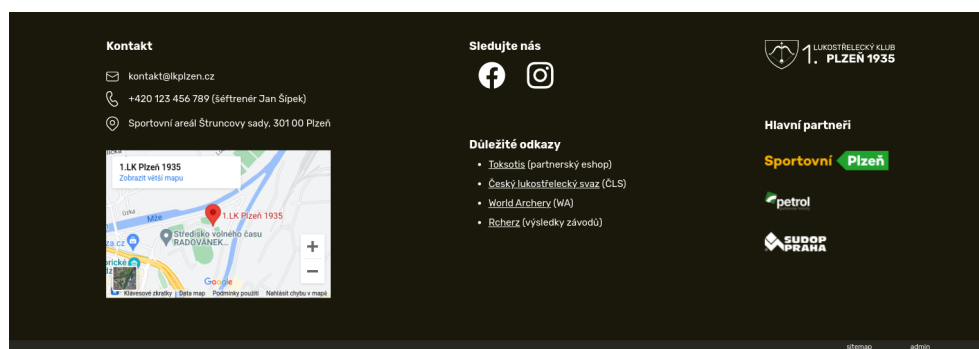
4. IMPLEMENTACE



Obrázek 4.5: Komponenta *HamburgerMenu*

Patička

Pro patičku jsem vytvořila komponentu *TheFooter*, která obsahuje kontaktní údaje, mapu s adresou střelnice, odkazy na sociální sítě, důležité odkazy a loga hlavních partnerů. Mapa je přidána pomocí elementu *iframe*, který jsem nechala vygenerovat aplikací Google Mapy [62]. Dále je pod hlavním obsahem úzký box, ve kterém jsou odkazy na administráční panel a soubor *sitemap*. Tato komponenta je zobrazena na obrázku 4.6.



Obrázek 4.6: Komponenta *TheFooter*

Profil

Komponenta *Profile* slouží pro zobrazení krátkého profilu na domovské stránce a na stránce *Klub*. Přebírá data o profilu z nadřazené komponenty a zobrazuje je do předem dané struktury.

Aktualita

Komponenta pro aktualitu se nazývá *Post*. Stejně jako předchozí komponenta přebírá data z nadřazené komponenty a zobrazuje je. Jelikož aktualita nutně nemusí mít titulní obrázek, zakomponovala jsem do kódu náhodně generovaná čísla, která využiji při výběru výchozího obrázku ze složky `assets/postPhotos`, pokud aktualita žádný nemá.

Úvod stránky

V úvodu každé stránky kromě stránky aktuality a profilu přidávám komponentu *Head*, která je zobrazena v obrázku 4.4. Skládá se z fotografie, která je v pozadí, a názvu stránky. Tyto atributy se vždy předávají z nadřazené komponenty.

4.2.2.2 Domovská stránka

Domovská stránka je složena z šesti sekcí. První sekcí je úvod stránky v podobě velké fotografie, motivačního textu a odkazu na další sekci. Sekce *Klub 1. LK Plzeň* obsahuje krátké informace o klubu s odkazy na více informací. Sekce *Lukostřelba* je podobná předchozí sekci, navíc obsahuje malé obrázky s názvy jednotlivých divizí luků. Čtvrtá sekce *Lidé v klubu* obsahuje profily lidí, kteří mají příznak `onHomepage`. Tyto profily jsou zobrazeny pomocí komponenty *Slider*. Následující sekce *Aktuality* (obrázek 4.7) se skládá z tří nejnovějších aktualit a odkazem na stránku *Závody*. Komponenta, která aktuality zobrazuje na domovské stránce, se nazývá *PostList*. Poslední sekcí na této stránce je sekce *Partneři*, která obsahuje komponentu *Partners*. V této komponentě jsou přidáni pouze hlavní partneři.



Obrázek 4.7: Sekce *Aktuality*

Profily a aktuality jsou dynamická data, které získávám z backendu pomocí Vuex. Pro profily jsem vytvořila jmenný prostor `profiles`, kde v `actions.js` posílám požadavek na endpoint `/profiles` s parametry, zda má profil atribut

`onHomepage` nastavený na `true`. Pro aktuality jsem vytvořila jmenný prostor `posts`, kde existuje funkce pro zaslání požadavku na endpoint `/posts` s parametry na řazení odpovědi sestupně, tedy od nejnovější aktuality.

V prvním odstavci této sekce jsem zmínila pár komponent, které v následujícím seznamu popíši:

Slider

Jedná se o komponentu, která zobrazuje profily, které získá z nadřazené komponenty. Vždy jsou zobrazeny jen tři (v menších zobrazeních dva nebo jeden) profily. Pro zobrazení slideshow jsem využila komponentu *Vueper Slides* [63], která je volně dostupná na platformě GitHub. Tato komponenta umožňuje automatické posouvání, posouvání šipkami a dotykem, responzivní design a mnoho dalších možností.

PostList

Tato komponenta má na starosti zobrazení tří nejnovějších aktualit a je zobrazena na obrázku 4.7. Z nadřazené komponenty získá data všech aktualit seřazených od nejnovější a pomocí JavaScriptové funkce `slice` zkrátí seznam na tři. K zobrazení jednotlivé aktuality využívá komponentu *Post*.

Partners

Komponenta *Partners* zobrazuje výčet předem daných hlavních partnerů. Každý partner má vždy logo a v něm odkaz na jeho webovou stránku.

4.2.2.3 Klub

Stránka *Klub* obsahuje kromě komponenty *Head*, kterou přidávám na každou stránku, dvě sekce. První z nich jsou jen statické informace o klubu s odkazy na jiné stránky. Druhá sekce *Lidé v klubu* obsahuje sekci, kde se může přesouvat mezi tabulkami se seznamy osob (členové klubu, trenéři, předsednictvo a ligová družstva). Každá tato tabulka je naplněná daty z backendu.

Pro získání dat z backendu jsem ve Vuex vytvořila jmenný prostor `people`, kde v souboru `actions.js` posílám požadavky na endpoint `/members` a na endpointy `/coaches`, `/committees` a `/league-teams` s parametry, které mi zajistí, že data přijdou i s daty člena a jeho profilu. V souboru `mutations.js` data zpracovávám do jiné podoby, které mi usnadní jejich následné zobrazení ve struktuře.

V sekci *Lidé v klubu* je také komponenta *ProfileSlide*, která podobně jako komponenta *Slider* využívá komponentu třetí strany *Vueper Slides*. Zobrazuje vždy pouze jeden profil.

4.2.2.4 Chci střílet z luku

Stránka *Chci střílet z luku* se skládá ze čtyř sekcí. První sekcí jsou otázky a odpovědi, které by měly pomoci osobě se zájmem o střelbu z luku. Tato sekce obsahuje statická data. Druhá sekce *Lukostřelba pro veřejnost*, obsahuje formulář a informace o lekcích pro veřejnost, která jsou získávána z backendu. Formulář je přidán pomocí komponenty *Form* a informace pomocí komponenty *CourseInfo*. Třetí sekce *Jak se stát členem klubu* je podobná té předchozí, navíc obsahuje výpis termínů a je zobrazena v obrázku 4.8. Poslední sekce obsahuje jen informace o možnosti firemních akcí, která jsou také získávána z backendu.

Jak se stát členem klubu

Zájemce o vstup do našeho klubu přijímáme výhradně prostřednictvím Kurzu lukostřelby pro začátečníky. Tyto kurzy pořádáme dva ročně, a to v dubnu, na začátku venkovní sezóny a v září, před začátkem halové sezóny. Ahoj byl jsem zde

Kurz lukostřelby pro začátečníky

Kurz je určen začátečníkům, kteří by chtěli začít s lukostřelbou a stát se členem klubu. Na prvním tréninku jsou vysvětleny základy techniky, jaké existují druhy luků a jakými lukostřeleckými disciplínami je možno se zabývat. S vypůjčeným vybavením si začátečník vyzkouší zhruba 40 výstřelů.

V následujících třech trénincích (každý po dvou hodinách) se vysvětluje základ lukostřelecké techniky. Po absolvování kurzu by měl mít každý účastník dostatečný přehled o tom, co sportovní lukostřelba obnáší a jestli je to opravdu to, čemu by rád věnoval svůj volný čas. Pokud ano, je možné pomocí přihlášky požádat o členství v klubu a začít trénovat pravidelně.

- cena kurzu je 1500 Kč
- v ceně kurzu je pronájem prostor lukostřelnice, trenérské zajištění a půjčovné materiálu (luk, šípy, chrániče)

Dostupné termíny:

První termín:	Další termíny:
8.7.2022 15:00 - 16:30	15.9.2022 15:00 - 16:00
22.7.2022 15:00 - 16:00	23.9.2022 15:00 - 16:00
29.7.2022 15:00 - 16:00	30.9.2022 15:00 - 16:00

Do kurzu se prosím přihlašuj pomocí formuláře. Pokud máš nějaký další dotaz ohledně kurzu lukostřelby pro začátečníky, kontaktuj:

Tadeáš Kalvas
 ✉ anjane007@gmail.com
 ☎ +420752863974

Přihlaš se na kurz lukostřelby pro začátečníky

Jméno a příjmení zájemce*

Věk zájemce*

Kontaktní e-mail*

Telefonní číslo

Vyber si kurz*

9.9.2022 15:00 - 16:00

Poznámky/zpráva

Souhlasím se zpracováním osobních údajů.

Odeslat přihlášku

Obrázek 4.8: Sekce *Jak se stát členem klubu*

Navíc je na stránce přidána komponenta *Modal*, což je modální okno, které se zobrazí, pokud uživatel odesle formulář s daty. Informace v tomto okně jsou jiné pro různé odpovědi ze serveru. O stav zobrazení se stará jmenný prostor *modals*, který obsahuje název okna, zprávu a status, zda se jedná o chybu. Pomocí akcí *showModal* a *hideModal* se může modální okno zobrazit a skrýt.

Data z backendu získávám pomocí jmenného prostoru *pages*, kde volám čtyři endpointy: */public-archery*, */start-membership*, */start-course* a */company-course*. O odeslání dat z formuláře se stará jmenný prostor *messages*, který má pro každý formulář jinou funkci. Data posílá na endpoint */message* a typ formuláře. Využívá jmenného prostoru *modals*, díky kterému zobrazuje odpovědi ze serveru.

V této sekci jsem zmínila komponenty, které v následujícím seznamu popíši:

Form

Pro každý formulář na webu je využita komponenta *Form*. Obsahuje formulářové prvky a tlačítko k odeslání. Pro každý formulář jsou některé prvky jiné nebo chybí. Při odeslání formuláře se nejdříve data validují, a až poté se zavolá příslušná funkce z nadřazené komponenty. Validaci podléhají pouze pole jméno a příjmení, věk, e-mail a zaškrťávací pole, kterým uživatel dává souhlas se zpracováním dat.

CourseInfo

Komponenta *CourseInfo* má na starosti zobrazování dat k jednotlivým sekcím stránky *Chci strážlet z luku*. Data dostává z nadřazené komponenty.

Modal

Pro zobrazení modálního okna existuje komponenta *Modal*. Veškeré zobrazené informace přebírá z Vuex.

4.2.2.5 Kontakt

Všechny důležité kontakty obsahuje stránka *Kontakt*. Je zde adresa střelnice, informace k adrese a mapa, která je stejně jako v patičce přidána pomocí prvku `iframe`. Dále stránka obsahuje kontaktní formulář, který je přidán pomocí komponenty *Form*. Následují kontakty na významné členy předsednictva, obecná e-mailová adresa, korespondenční adresa klubu, bankovní spojení, IČ a právní informace. Všechny informace jsou zde přidány staticky.

4.2.2.6 Pro členy

Na stránce *Pro členy* se nachází informace, které jsou užitečné členům klubu. Jsou zde kalendáře pro jednotlivé věkové divize. Kalendář je vložen pomocí vygenerovaného `iframe` z aplikace Google Kalendář. Kalendář se dá také udělat vlastní se získáváním dat z Google Calendar API, ale podobný seznam událostí, jako je v současném řešení, by bylo náročné, proto jsem se rozhodla pro ponechání `iframe`, se kterým je klient spokojen a umožňuje například tisk kalendáře nebo přidání kalendáře do svého kalendáře v aplikaci Google Kalendář.

Další sekce obsahuje informace o soustředění, které jsou získávány z backendu funkcí ve Vuex, kterou jsem přidala do jmenného prostoru `pages`.

Poslední sekci jsou seznamy důležitých dokumentů a odkazů. Tyto seznamy jsou získány z backendu pod endpointem `for-member`. Důležité dokumenty jsou soubory různých typů, které jsou dostupné pod URL backendu. Odkazy se skládají ze tří částí: logo, název, který se zobrazí na stránce, a odkaz na webovou stránku.

4.2.2.7 Ostatní stránky

Stránky pro profil a aktualitu jsou dynamické. Na základě řetězce slug v adrese posílají požadavek přes Vuex na jeden konkrétní profil nebo aktualitu. Také využívají modul Markdownit.

Stránka *Závody* využívá pro zobrazení aktualit komponentu *Post*. Dále obsahuje sekce *Naše úspěchy* a *Klubové rekordy*, které jsou zobrazené v tabulkách.

Většina ostatních stránek obsahuje statická data, nebo jen získávají data z backendu a zobrazují je do připravené struktury. Také využívají již vytvořené komponenty, které jsou popsány výše.

Testování a nasazení

Po implementaci aplikace následuje její otestování a poté i nasazení. V této kapitole vysvětlím, jaké typy testů jsem využila a jaké výsledky přineslo testování prezentační části. Po otestování jsem aplikaci nasadila a popíši zde, jaké platformy jsem využila, z jakého důvodu a jakým způsobem probíhá nasazení aplikace na jednotlivých platformách. Pro opětovné sestavení prezentační části poté, co proběhne změna obsahu, jsem využila webhook, jehož nastavení jsem popsala na konci sekce o nasazení.

V poslední části této kapitoly se podívám na možná rozšíření a vylepšení aplikace, které vyplynuly při průběhu návrhu a implementace aplikace.

5.1 Testování

Testování proběhlo pouze na prezentační části, protože systém Strapi použitý v administrační části je již v produkci, tudíž by měl být otestovaný. Přidané funkce v administrační části byly otestovány manuálně v průběhu implementace. Prezentační část byla otestována uživatelskými a akceptačními testy.

5.1.1 Uživatelské testy

Uživatelské testy poskytují informace o tom, zda uživatelé jsou schopni splnit jejich cíle a jak moc jsou s postupem pro splnění spokojeni. Díky zpětné vazbě od uživatelů je možné vylepšit uživatelské rozhraní a další aspekty aplikace.

Web je ve velké části zaměřen na tři hlavní skupiny osob:

- osoby se zájmem o lukostřelbu a rodiče, které chtějí svému dítěti zajistit zájmový kroužek,
- členové klubu,
- sponzoři, kteří chtějí vidět, zda klub funguje.

Pro každou skupinu osob jsem vytvořila testovací scénáře, které jsou složené z testovacích případů. Scénáře se zaměřují na nejdůležitější prvky webu a jejich funkčnost. Pro testování jsem vybrala osoby, které mají k dané skupině blízko. S osobou jsem prošla scénář a popíšu, jak zvládla splnit cíl a jaké připomínky a nápady k dané sekci měla.

5.1.1.1 Osoba se zájmem o lukostřelbu

Pro tuto skupinu jsem vybrala osobu, která by si chtěla lukostřelbu zkusit, ale nenašla jsem osobu, která by byla rodičem. Popsala jsem tedy situaci, aby se osoba mohla vcítit do role a prošla jsem s ní následující scénář:

1. Jsi rodič dítěte, kterému je osm let a chtělo by dělat lukostřelbu. Proto ses rozhodl ho přihlásit do nejbližšího lukostřeleckého klubu. Našel jsi webovou stránku 1. lukostřeleckého klubu Plzeň 1935. Kde bys hledal na této stránce informace o tom, jak můžeš přihlásit své dítě na lukostřelbu?
2. Dozvěděl ses, že ke členství v klubu je nutnost absolvovat kurz lukostřelby pro začátečníky, na který se dá přihlásit pomocí přiloženého formuláře. Vyplň ho a odešli.

První případ začal na hlavní stránce. Osoba přešla pomocí karty *Chci střílet z luku* na tuto stránku. Přečetla si sekci s otázkami a odpověďmi, ze které následně otevřela odkazy na sekci *Lukostřelba pro veřejnost* a sekci *Jak začít s lukostřelbou v nových kartách*. Z toho se dá vyčíst, že osoba předpokládala, že informace jsou na jiných stránkách. Dále přešla na kartu s otevřenou sekci *Lukostřelba pro veřejnost*, u které si řekla, že je to spíše nějaký jednodenní kurz pro dospělé. Následně se přesunula na kartu s otevřenou sekci *Jak se stát členem klubu*, kde našla požadované informace.

Ze zpětné vazby osoby k tomuto případu jsem se dozvěděla, že otázka v druhém bodě sekce s otázkami a odpověďmi je moc dlouhá a osoba ji nedočetla do konce, proto se nejdříve podívala na sekci *Lukostřelba pro veřejnost*.

Druhý případ začal na místě, kde skončila osoba v prvním případě. Formulář vyplnila dle očekávání a úspěšně ho odeslala. Následně zkusila vyplnit také špatná data a ocenila, že je provedena validace. Co ji vadilo na formuláři, byla kolonka *Jméno a příjmení zákonného zástupce*, která se zobrazí až poté, co se vyplní věk menší než 18 let. Kolonku by ve formuláři nechala zobrazenou.

5.1.1.2 Člen klubu

Osoba, která byla vybrána pro tuto skupinu, je současným členem klubu. Pro otestování jsem vybrala následující případy:

1. Jsi člen klubu a na stránkách hledáš informace o nadcházejících závodech. Kde bys je hledal?

2. Na tréninku jsi se dozvěděl o dokumentu zakázaných léků, který je prý někde na stránkách ke stažení. Kde bys ho hledal?

Test začal na domovské stránce a v obou případech osoba klikla na rozbalovací menu. V prvním případě nejdříve hledala nadcházející závody na stránce *Závody* v sekci *Aktuality*. Následně však našla požadované kalendáře pomocí položky v menu *Kalendáře*. Kalendář má dle jejího názoru dobrou funkčnost, ale nevypadá dobře. Této skutečnosti jsem si vědoma, ale vygenerovaný `iframe` se nedá upravovat.

Ve druhém případě scénáře osoba hledala dokument na stránce *Lukostřelba*. Dokument však našla po kliknutí na položku *Dokumenty a odkazy* v rozbalovacím menu.

Web má dle osoby dobrou strukturu a ocenil stránku *Chci strážet z luku*, která na stávajícím webu chyběla.

5.1.1.3 Sponzor

Pro tuto skupinu nejsem schopna vybrat správnou osobu s těmito zájmy, proto jsem v následujícím scénáři popsala situaci, aby se vybraná osoba mohla vcítit do role.

1. Jsi potencionální sponzor a chtěl bys podpořit nějaký lukostřelecký klub. Náhodou jsi narazil na webové stránky 1. lukostřeleckého klubu Plzeň 1935 a chtěl bys vědět, zda je klub aktivní. Co bys hledal a kde?
2. Poté bys chtěl zjistit, jací další sponzoři klub podporují a rozhodl ses, že klub podpoříš. Najdi kontakt, na který by ses obrátil.

První případ začal na domovské stránce. Vybraná osoba se přesunula do karty *O klubu*, kde hledala informace o klubu. Následně se pomocí rozbalovacího menu přesunula do sekce *Závody*, kde si prohlédla aktuality a jednu z nich otevřela, a poté si prohlédla úspěchy klubu. Tyto informace by jí dle jejího názoru stačili k tomu, že je klub aktivní.

V druhém případě test začal na stejném místě, kde skončil první případ. Osoba klikla na kartu *Partneři*, kde si prohlédla všechny partnery, ale nevěšila si motivačního nápisu, který je na konci stránky. Proto se přesunula do karty *Kontakty*, kde našla e-mail, na který by se obrátila. Na dotaz, zda by použila formulář, odpověděla, že ne, protože je podle ní na obecné dotazy k fungování klubu.

5.1.2 Akceptační testy

Nejdříve jsem s klientem prošla prezentační část, její jednotlivé stránky a sekce. Dle jeho názoru stránky obsahují vše, co by měli obsahovat a jsou pěkně

zpracované. Vytkl mi pár drobností, které spíše souvisely s vizuální stránkou. Tyto drobnosti jsem upravila do současné podoby.

Dále jsem klientovi ukázala administrační část, popsala její fungování a kde se typy obsahu nachází na prezentační části. Vysvětlila jsem způsob přidání obsahu a médií, který klientovi přijde velmi jednoduchý.

Celkově byl klient s oběma částmi velmi spokojený, webová aplikace splnila většinu jeho požadavků a v následující práci na prezentační části dodá texty a fotografie, které budou ve výsledném ostrém nasazení.

5.1.3 Výsledky testování

Testování až na pár případů dopadlo dle očekávání. Všichni účastníci byli schopni splnit zadané úlohy v krátkém čase. Dle zpětného názoru vybraných osob pro testování a klienta jsem předělala části, které nebyly dle očekávání.

5.2 Nasazení

Aplikaci jsem nasadila na dvě různé platformy. V následujících sekcích popíši, jaké platformy byly využity a jakým způsobem probíhá nasazení na ně. Následně popíši, jak jsem využila webhook.

5.2.1 Backend

Pro nasazení systému Strapi jsem využila platformu Heroku [64]. Jedná se o cloudovou platformu, která podporuje několik programovacích jazyků. Mezi nimi je také Node.js, na kterém je Strapi postaveno. Pro nasazení je nutné mít projekt verzovaný systémem Git.

Platformu Heroku jsem vybrala proto, že má velmi jednoduchou možnost nasazení aplikace a nabízí bezplatný základní plán, který se může na ostrý provoz vylepšit na další placené plány.

Nasazení probíhá pomocí Heroku CLI. Po jeho stažení je nutností se přihlásit pomocí účtu na Heroku a přidat pomocí CLI nový vzdálený repozitář aplikace, která lze vytvořit na webovém rozhraní Heroku.

Pro databázi jsem využila doplněk pro PostgreSQL. Pro jeho nastavení je nutné vytvořit konfigurační soubory. Přidala jsem proto do administrační části do složky `/config/env/production` soubor `database.js` pro konfiguraci databázi v produkci a soubor `server.js` pro poskytnutí údajů o veřejné Heroku doméně systému Strapi. Pro konfiguraci databáze jsem ještě využila balíček `pg-connection-string`, který zjednodušuje parsování databázové konfigurace poskytované platformou Heroku.

Po nastavení projektu jsem všechny změny přidala do verzovacího systému Git a pomocí Heroku CLI nasadila na cloud.

5.2.2 Frontend

Jako hosting pro staticky generovaný frontend jsem zvolila Vercel [65]. Jde o platformu pro frontendové frameworky a statické stránky. Stará se o nasazení, automatické škálování a poskytování personalizovaného obsahu. Nasazení je velmi snadné a pro základní použití je platforma bezplatná. Stejně jako u Heroku, je nutnost mít projekt verzovaný systémem Git.

Pro nasazení je možné do platformy Vercel importovat existující repozitář z jedné ze tří platforem (GitHub, GitLab a Bitbucket). Dále se nastaví informace k automatickému nasazení, jako je například příkaz pro sestavení projektu, který jsem v nastavení změnila. Následně se může projekt nasadit a je volně přístupný pod vygenerovanou doménou, která se vždy může změnit na jinou.

Když je již projekt nasazený, všechny následující změny v hlavní větvi verzovacího systému znovu spustí proces sestavení.

5.2.3 Webhook

Protože je frontend staticky generovaný, změna obsahu na backendu se nepropíše, dokud se frontend znovu nesestaví. K informaci o změně obsahu předá frontend webhook, který jsem nastavila následovně: Ve Vercelu jsem si v nastavení projektu nechala vygenerovat *Deploy Hook*, což je unikátní URL adresa, která umožňuje spustit proces sestavení frontendu. Poté jsem ve Strapi v nastavení vytvořila nový webhook, kterému jsem dala jméno a vygenerovanou URL adresu, a nastavila jsem události, při kterých se webhook spustí. Tyto události jsou vytvoření, úprava a smazání obsahu, jeho publikace a zrušení publikace.

5.3 Budoucnost vývoje webové aplikace

Webovou aplikaci je možné neustále zdokonalovat a rozšiřovat. Díky zvoleným technologiím není problém do obou částí cokoli přidat. Pokud by klient chtěl například novou stránku, jednoduše se přidá nový typ obsahu do Strapi a na prezentační části se musí vytvořit požadovaná stránka.

V průběhu návrhu a implementace vyplynuly následující vylepšení a rozšíření. V části funkcionalit by se jednalo o:

- implementaci možnosti nahrání dat ze souboru do daného typu obsahu v administrační části,
- automatické odstranění termínu kurzu lukostřelby pro začátečníky po jeho naplnění,
- jiné řešení rezervace lekcí a kurzů než pomocí automatických e-mailů (automatické e-maily byl však klientův požadavek),

5. TESTOVÁNÍ A NASAZENÍ

- správu SEO.

V prezentační části je možné vylepšit design a implementovat přívětivější kalendář s použitím Google Calendar API, který by byl stylově ekvivalentní designu webu.

Závěr

Cílem této práce bylo vytvoření nové webové aplikace pro 1. lukostřelecký klub Plzeň 1935, která splňuje požadavky klienta, je uživatelsky přívětivá a prezentační část má modernější design. V první kapitole byla popsána motivace, cíle práce a její struktura.

V druhé kapitole bylo zanalyzováno současné řešení, jeho nevyhovující vzhled a technické údaje. Poté byl proveden sběr uživatelských požadavků a řešerše webů s podobnou tematikou, ze které vyšly tři webové stránky, které byly podrobněji popsány. Dále byly popsány jednotlivé možnosti řešení nového webu. Tato sekce se zaměřila na dostupné systémy pro správu obsahu, jejich výhody a nevýhody a porovnání s možným vlastním řešením. Byla také provedena řešerše možností, jakými lze vytvořit prezentační část. Na konci této kapitoly byla vybrána možnost headless CMS pro administrační část a pro prezentační část byla vybrána možnost vývoje přes framework, který podporuje SSG.

Třetí kapitola byla zaměřena na návrh obou částí aplikace. Pro administrační část byl vybrán systém Strapi. Pro prezentační část byly vytvořeny wireframy, které byly konzultovány s klientem, dále byl proveden výběr a jeho odůvodnění technologií pro implementaci prezentační části. Nakonec byla popsána architektura aplikace a proveden výběr API pro distribuci dat.

Ve čtvrté kapitole byl proveden popis implementace aplikace. Byla popsána konfigurace administrační části, vytvořené typy obsahu, přidané funkcionality a přizpůsobení administrátorského panelu. Následně byl vysvětlen souborový systém, konfigurace a vývoj prezentační části. Vývoj se zaměřil na jednotlivé stránky a použité komponenty.

Pátá kapitola se zaměřila na testování aplikace a její následné nasazení. Pro testování byly vybrány uživatelské a akceptační testy, které byly následně provedeny a výsledky v práci popsány. Poté bylo provedeno nasazení na cloudové platformy, které byly popsány v této kapitole. Poslední část kapitoly popsala možná rozšíření a vylepšení aplikace.

Výsledná aplikace prošla popsánými testy a klient je s výsledkem spoko-

ZÁVĚR

jený. Aplikace splnila jeho požadavky a po dodání obsahu a fotografií bude nasazena do ostrého provozu pro propagaci 1. lukostřeleckého klubu Plzeň 1935.

Bibliografie

1. LUKOSTŘELECKÝ KLUB PLZEŇ 1935, Z. S. *Web 1. LK Plzeň 1935* [online]. 2016 [cit. 2022-04-24]. Dostupné z: <http://lkplzen.cz/>.
2. GOOGLE LLC. *Google Kalendář* [soft.]. 2009 [cit. 2022-04-24]. Dostupné z: <https://www.google.com/intl/cs/calendar/about/>.
3. ŠÍPEK, Jan. *Konverzace o nové webové stránce* [ústní sdělení]. 2021.
4. STATCOUNTER. *Browser Market Share in Europe* [online]. 2022 [cit. 2022-04-08]. Dostupné z: <https://gs.statcounter.com/browser-market-share/all/europe/#monthly-202110-202202-bar>.
5. LUKOSTŘELECKÝ ODDÍL TJ OPAVA, Z.S. *Web LO TJ Opava* [online]. 2017 [cit. 2022-04-24]. Dostupné z: <http://www.lukostrelbaopava.cz/>.
6. BOJCUN, Miroslav; ZEMAN, Viktor. *Rcherz* [online]. 2012 [cit. 2022-04-24]. Dostupné z: <https://frontend.rcherz.com/>.
7. LUKOSTŘELECKÝ ODDÍL TJ OPAVA, Z.S. *Partneři na webu LO TJ Opava* [online]. 2017 [cit. 2022-04-24]. Dostupné z: <http://www.lukostrelbaopava.cz/?e=partneri>.
8. LO TJ BRNO A LK BRNO 05. *Web Lukostřelba Brno* [online]. 2015 [cit. 2022-04-24]. Dostupné z: <http://www.lukostrelbabrno.cz/>.
9. REDHILL ARCHERS. *Web klubu Redhill Archers* [online]. 2021 [cit. 2022-04-24]. Dostupné z: <https://redhillarchers.co.uk/>.
10. BARKER, Deane. *Web Content Management: Systems, Features, and Best Practices*. 1. vyd. O'Reilly Media, Inc., 2016. ISBN 978-1-491-90812-9.
11. PRIYA. *What is a content management system (CMS)? 8 CMS you need to know* [online]. 2022 [cit. 2022-03-04]. Dostupné z: <https://themeisle.com/blog/what-is-a-content-management-system-cms/>.

12. BOHAN, Donna-Marie. *What is a content management system and how does it work?* [Online]. 2019 [cit. 2022-03-05]. Dostupné z: <https://www.bloomreach.com/en/blog/2019/content-management-system>.
13. FJORGE. *Coupled, decoupled, or headless CMS: What's the difference?* [Online]. 2019 [cit. 2022-03-05]. Dostupné z: <https://fjorgedigital.com/insights/blog/what-is-the-difference-between-a-coupled-decoupled-headless-cms/>.
14. BRIGHTSPOT. *Headless CMS pros and cons: Benefits of headless CMS and other CMS types* [online]. 2018 [cit. 2022-03-05]. Dostupné z: <https://www.brightspot.com/learn/articles/decoupled-cms-and-headless-cms-platforms>.
15. MESHEN, Laura. *CMS Architecture: Traditional, decoupled and headless* [online]. 2021 [cit. 2022-03-05]. Dostupné z: <https://www.acromedia.com/article/cms-architecture-traditional-decoupled-headless>.
16. ZUCREW. *Coupled vs. decoupled and headless CMS Architectures* [online]. 2021 [cit. 2022-03-05]. Dostupné z: <https://zu.com/articles/coupled-vs-decoupled-headless-cms-architectures>.
17. AUTOMATTIC INC. *WordPress* [soft.]. 2003 [cit. 2022-03-09]. Dostupné z: <https://wordpress.org/>.
18. W3TECHS. *Usage statistics of content management systems* [online]. 2022 [cit. 2022-03-09]. Dostupné z: https://w3techs.com/technologies/overview/content_management.
19. JOHNSON, Desiree. *Joomla vs. WordPress vs. Drupal* [online]. 2020 [cit. 2022-03-09]. Dostupné z: <https://www.bluehost.com/resources/joomla-vs-wordpress-vs-drupal/>.
20. FITZGERALD, Anna. *Drupal vs WordPress vs Joomla vs HubSpot: Which is the right CMS for you?* [Online]. 2021 [cit. 2022-03-09]. Dostupné z: <https://blog.hubspot.com/website/wordpress-vs-joomla-vs-drupal>.
21. KOŘOUSKOVÁ, Barbora. *Srovnání redakčních systémů: WordPress, Joomla, Nebo Drupal?* [Online]. 2021 [cit. 2022-03-09]. Dostupné z: <https://www.rascasone.com/cs/blog/srovnani-cms-wordpress-joomla-drupal>.
22. BUYTAERT, Dries. *Drupal* [soft.]. 2001 [cit. 2022-03-09]. Dostupné z: <https://www.drupal.org/>.
23. OPEN SOURCE MATTERS INC. *Joomla!* [Soft.]. 2005 [cit. 2022-03-09]. Dostupné z: <https://www.joomla.org/>.
24. ŠTRÁFELDA, Jan. *Co je headless CMS* [online]. 2020 [cit. 2022-03-09]. Dostupné z: <https://www.strafelda.cz/headless-cms>.

25. STEPNOV, Eugene. *Best Headless CMS in 2022* [online]. 2022 [cit. 2022-03-10]. Dostupné z: <https://flatlogic.com/blog/best-headless-cms/>.
26. CONTENTFUL GMBH. *Contentful* [soft.]. 2013 [cit. 2022-03-10]. Dostupné z: <https://www.contentful.com/>.
27. KOTHARI, Abhishek. *11 Headless CMS to Consider for Modern Application* [online]. 2022 [cit. 2022-03-10]. Dostupné z: <https://geekflare.com/headless-cms/>.
28. GREEN, Dave. *Contentful* [online]. 2021 [cit. 2022-03-10]. Dostupné z: <https://bejamas.io/discovery/headless-cms/contentful/>.
29. STRAPI SOLUTIONS SAS. *Strapi* [soft.]. 2015 [cit. 2022-03-10]. Dostupné z: <https://strapi.io/>.
30. MARTINEZ, Gerald. *Strapi* [online]. 2021 [cit. 2022-03-10]. Dostupné z: <https://bejamas.io/discovery/headless-cms/strapi/>.
31. NETLIFY, INC. *Netlify CMS* [soft.]. 2016 [cit. 2022-03-10]. Dostupné z: <https://www.netlifycms.org/>.
32. ADERALDO, Bruno. *NetlifyCMS* [online]. 2021 [cit. 2022-03-10]. Dostupné z: <https://bejamas.io/discovery/headless-cms/netlifycms/>.
33. PALAS, Petr. *How I built a CMS, and why you shouldn't* [online]. 2017 [cit. 2022-04-08]. Dostupné z: <https://hackernoon.com/how-i-built-a-cms-and-why-you-shouldnt-daff6042413a>.
34. MDN CONTRIBUTORS. *Getting started with HTML* [online]. 2022 [cit. 2022-05-04]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started.
35. THE WORLD WIDE WEB CONSORTIUM. *CSS Standarts* [online]. 1998 [cit. 2022-05-04]. Dostupné z: <https://www.w3.org/Style/CSS/>.
36. MDN CONTRIBUTORS. *What is CSS?* [Online]. 2022 [cit. 2022-05-04]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS.
37. MDN CONTRIBUTORS. *CSS preprocessor* [online]. 2021 [cit. 2022-05-04]. Dostupné z: https://developer.mozilla.org/en-US/docs/Glossary/CSS_preprocessor.
38. MDN CONTRIBUTORS. *JavaScript* [online]. 2022 [cit. 2022-05-04]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Glossary/JavaScript>.
39. MDN CONTRIBUTORS. *Introduction to client-side frameworks* [online]. 2022 [cit. 2022-05-04]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Introduction.

40. META PLATFORMS IRELAND LIMITED. *React* [soft.]. 2013 [cit. 2022-05-04]. Dostupné z: <https://reactjs.org/>.
41. SAKOVICH, Natallia. *Top Most Popular Frontend Frameworks 2022* [online]. 2021 [cit. 2022-05-04]. Dostupné z: <https://www.sam-solutions.com/blog/best-frontend-framework/>.
42. TECHNOSTACKS. *Top Frontend Frameworks of 2022 for Web Development* [online]. 2021 [cit. 2022-05-04]. Dostupné z: <https://technostacks.com/blog/best-frontend-frameworks/>.
43. GOOGLE LLC. *Angular* [soft.]. 2010 [cit. 2022-05-04]. Dostupné z: <https://angular.io/>.
44. YOU, Evan. *Vue.js* [soft.]. 2014 [cit. 2022-05-04]. Dostupné z: <https://vuejs.org/>.
45. MAINA, Moses. *Client Side Rendering vs. Server Side Rendering vs. Static Site Generation* [online]. 2021 [cit. 2022-05-04]. Dostupné z: <https://www.section.io/engineering-education/client-side-rendering-vs-server-side-rendering-vs-static-site-generation/>.
46. TEAM AT SEARCH ENGINE LAND. *What Is SEO / Search Engine Optimization?* [Online]. 2010 [cit. 2022-05-04]. Dostupné z: <https://searchengineland.com/guide/what-is-seo>.
47. NIELSEN, Emily. *What is the Difference Between SPAs, SSGs, and SSR?* [Online]. 2021 [cit. 2022-05-04]. Dostupné z: <https://graphcms.com/blog/difference-spa-ssg-ssr>.
48. CLOUDINARY LTD. *Cloudinary* [soft.]. 2012 [cit. 2022-05-04]. Dostupné z: <https://cloudinary.com/>.
49. THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. *PostgreSQL* [soft.]. 1996 [cit. 2022-05-04]. Dostupné z: <https://www.postgresql.org/about/>.
50. VUE.JS CORE TEAM. *Vuex* [online]. 2016 [cit. 2022-05-08]. Dostupné z: <https://vuex.vuejs.org/>.
51. CATLIN, Hampton. *Syntactically Awesome Style Sheets* [online]. 2006 [cit. 2022-05-07]. Dostupné z: <https://sass-lang.com/>.
52. HANNAH, Jaye. *What Exactly Is Wireframing? A Comprehensive Guide* [online]. 2021 [cit. 2022-05-04]. Dostupné z: <https://careerfoundry.com/en/blog/ux-design/what-is-a-wireframe-guide/>.
53. DUBINSKÁ, Lída. *Co je wireframe webu, proč ho potřebujete a jak ho vytvořit?* [Online]. 2021 [cit. 2022-05-04]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-wireframe-predstavujeme-5-duvodu-proc-je-pro-klienty-drateny-model-dulezity>.
54. WALLACE, Evan; FIELD, Dylan. *Figma* [online]. 2015 [cit. 2022-05-04]. Dostupné z: <https://www.figma.com/>.

-
55. TEAM WEBDESIGN INSPIRATION. *Webdesign Inspiration* [online]. 2021 [cit. 2022-05-03]. Dostupné z: <https://www.webdesign-inspiration.com/>.
 56. TUTORIALSPPOINT. *MVC Framework - Introduction* [online]. 2015 [cit. 2022-05-04]. Dostupné z: https://www.tutorialspoint.com/mvc-framework/mvc_framework_introduction.htm.
 57. SENDGRID TEAM. *What's a Webhook?* [Online]. 2014 [cit. 2022-05-09]. Dostupné z: <https://sendgrid.com/blog/whats-webhook/>.
 58. IBM CLOUD EDUCATION. *REST APIs* [online]. 2021 [cit. 2022-05-04]. Dostupné z: <https://www.ibm.com/cloud/learn/rest-apis>.
 59. NPM, INC. *npm* [online]. 2010 [cit. 2022-05-06]. Dostupné z: <https://www.npmjs.com/>.
 60. SENDINBLUE. *Sendinblue* [online]. 2007 [cit. 2022-05-06]. Dostupné z: <https://www.sendinblue.com/>.
 61. NUXT COMMUNITY. *Modul Markdownit* [online]. 2017 [cit. 2022-05-09]. Dostupné z: <https://www.npmjs.com/package/@nuxtjs/markdownit>.
 62. GOOGLE LLC. *Google Mapy* [soft.]. 2005 [cit. 2022-05-08]. Dostupné z: <https://www.google.com/maps>.
 63. ANDRE, Antoni. *Vueper Slides* [soft.]. 2018 [cit. 2022-05-10]. Dostupné z: <https://github.com/antoniandre/vueper-slides>.
 64. SALESFORCE.COM, INC. *Heroku* [online]. 2007 [cit. 2022-05-11]. Dostupné z: <https://www.heroku.com/>.
 65. VERCEL INC. *Vercel* [online]. 2015 [cit. 2022-05-11]. Dostupné z: <https://vercel.com/>.

Seznam použitých zkratk

API Application Programming Interface

CMS Content Management System

CSR Client-Side Rendering

CSS Cascading Style Sheet

CSV Comma-Separed Values

DOM Document Object Model

FAQ Frequently Asked Questions

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

IoT Internet of Things

MVC Model-View-Controller

PHP Hypertext Preprocessor

PNG Portable Network Graphics

REST Representational State Transfer

SEO Search Engine Optimalization

SMTP Simple Mail Transfer Protocol

SPA Single-Page Application

A. SEZNAM POUŽITÝCH ZKRATEK

SSG Static Site Generation

SSR Server-Side Rendering

SQL Structured Query Language

URI Uniform Resource Identifier

URL Uniform Resource Locator

XML Extensible Markup Language

Obsah přiloženého média

readme.md.....	stručný popis obsahu SD karty
links.md.....	výpis odkazů
wireframes.....	všechny použité wireframy při návrhu
src	
├── impl.....	zdrojové kódy implementace
│ ├── frontend.....	zdrojové kódy prezentační části
│ └── backend.....	zdrojové kódy administrační části
└── thesis.....	zdrojová forma práce ve formátu \LaTeX
text.....	text práce
└── thesis.pdf.....	text práce ve formátu PDF