

Střední průmyslová škola Třebíč

Maturitní práce

přepracování webové aplikace chovu zvířat

Profilová část maturitní zkoušky

Studijní obor: Informační technologie

Třída: ITA4

Školní rok: 2024/2025 Daniel Křížek

Obsah obrázku text, snímek obrazovky, Písmo, dopis

Popis byl vytvořen automaticky

ABSTRAKT

Tvorba maturitní práce je jedním z velmi klíčových momentů při studiu. Kvalita zpracování její formální části je pak jedním z nejdůležitějších kritérií při jejím hodnocení. Cíl této práce je popsat jednotlivé kroky během tohoto procesu, doporučit postupy a vytvořit šablonu, která usnadní celý proces.

KLÍČOVÁ SLOVA

maturitní práce, šablona

ABSTRACT

The creation of a graduation thesis is one of the most crucial moments during studies. The quality of the processing of its formal part is then one of the most important criteria in its evaluation. The aim of this work is to describe the individual steps during this process, recommend procedures and create a template that will facilitate the entire process.

KEYWORDS

graduation thesis, template

PODĚKOVÁNÍ

Děkuji Mgr. Petru Novotnému za cenné připomínky a rady, které mi poskytl při vypracování maturitní práce.

V Třebíči dne 26. března 2025 podpis autora

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl v ní všechny prameny, literaturu a ostatní zdroje, které jsem použil.

V Třebíči dne 26. března 2025

podpis autora

Obsah

[Úvod 7](#_Toc193895355)

[1 Teoretická část 8](#_Toc193895356)

[1.1 Webové stránky 8](#_Toc193895357)

[1.1.1 Tvorba webových stránek 8](#_Toc193895358)

[1.2 Hosting 8](#_Toc193895359)

[1.2.1 Doména 8](#_Toc193895360)

[1.3 Frontend 9](#_Toc193895361)

[1.3.1 UX 9](#_Toc193895362)

[1.3.2 UI 9](#_Toc193895363)

[1.3.3 Frontend technologie 9](#_Toc193895364)

[1.3.4 HTML 9](#_Toc193895365)

[1.3.5 CSS 10](#_Toc193895366)

[1.3.6 JavaScript 10](#_Toc193895367)

[1.3.7 JavaScript Framework 10](#_Toc193895368)

[1.4 Backend 10](#_Toc193895369)

[1.4.1 Technologie backendu 10](#_Toc193895370)

[1.4.2 Server 11](#_Toc193895371)

[1.4.3 API 11](#_Toc193895372)

[1.4.4 Databáze 11](#_Toc193895373)

[1.4.5 Databáze relační 12](#_Toc193895374)

[1.4.6 NoSQL databáze 12](#_Toc193895375)

[1.4.7 MongoDB 13](#_Toc193895376)

[1.4.8 PHP 13](#_Toc193895377)

[1.4.9 Node.js 13](#_Toc193895378)

[1.5 Frontend VS Backend 14](#_Toc193895379)

[2 Praktická část 15](#_Toc193895380)

[2.1 Výběr projektu 15](#_Toc193895381)

[2.2 Výběr programovacích jazyků 15](#_Toc193895382)

[2.3 Výběr vývojových nástrojů 16](#_Toc193895383)

[2.3.1 Výběr editoru kódu 16](#_Toc193895384)

[2.3.2 Výběr verzovacího systému 16](#_Toc193895385)

[2.3.3 Výběr nástroje pro design 17](#_Toc193895386)

[2.3.4 Výběr lokálního serveru 18](#_Toc193895387)

[2.3.5 Výběr editoru databází 19](#_Toc193895388)

[2.4 Tvorba webové stránky 19](#_Toc193895389)

[2.4.1 Návrh 20](#_Toc193895390)

[21](#_Toc193895391)

[2.4.2 Tvorba navigačního panelu (navbaru) 21](#_Toc193895392)

[2.4.3 Tvorba formuláře pro přihlášení a registraci 23](#_Toc193895393)

[2.4.4 Tvorba měniče jazyku 25](#_Toc193895394)

[2.4.5 Tvorba úvodní stránky 26](#_Toc193895395)

[2.5 Přechod do Reactu 27](#_Toc193895396)

[2.6 Odmítnutí klienta 27](#_Toc193895397)

[Závěr 28](#_Toc193895398)

[Seznam použitých symbolů a zkratek 30](#_Toc193895399)

[Seznam použitých zdrojů 31](#_Toc193895400)

[Seznam obrázků 34](#_Toc193895401)

[Seznam tabulek 35](#_Toc193895402)

[Seznam příloh 36](#_Toc193895403)

Úvod

Tento dokument je zaměřen na zdokumentování a podrobný popis celého procesu tvorby webové stránky pro chovatelskou stanici Garfield’s Baby. Tato chovatelská stanice se zaměřuje na chov mainské mývalí kočky. Cílem tohoto projektu bylo vytvoření moderní, přehledné a funkční webové stránky, která bude sloužit jako zdroj informací pro současné, ale i potencionálně budoucí vlastníky mainské mývalí kočky. Webová stránka měla za úkol nejen informovat o chovatelské stanici, ale i předat přehled o aktuální dostupnosti koček, kocourů a koťat. Na webové stránce lze také najít veškeré záznamy o všech vrzích chovatelské stanice. Dokumentace také zahrnuje všechny použité technologie, struktury webové stránky a designu. Zahrnut je i přehled o testování a SEO. Při tvorbě webové stránky byl kladen veliký důraz na responzivní design, který umožní pohodlný zážitek pro uživatele desktopových, ale i mobilních zařízení.

Motivace k projektu, citové zabarvení…

# Teoretická část

## Webové stránky

Webové stránky jsou nedílnou součástí digitálního světa, které jsou publikovány na internetu a mají mnoho využití. Webové stránky mohou sloužit jako zdroj informací, nabídka online služeb a prodej produktů (e-shopy), sdílení zážitků a různých poznatků (blogy) a mnoho dalších.

### Tvorba webových stránek

Při tvorbě webových stránek je potřeba zvážit několik různých věcí. Úplně každá webová stránka musí být na internet nějak vložená a k tomu slouží webhosting. Dále je potřeba upřesnit si, jakou má mít funkci samotná webová stránka. Webová stránka může být portfolio, e-shop, blog, diskuzní fórum nebo třeba sociální síť. Při vývoji webových stránek se používají 3 hlavní vrsty, frontend, backend a databáze.

## Hosting

Webhosting je místo na internetu, kde jsou uložené soubory webu a e‑mailové schránky. Zprostředkovává jej nepřetržitě běžící počítač se speciálním softwarem (server). Webhosting potřebuje ke správnému fungování doménu, která je na něj nasměrovaná. [15]

### Doména

Doména je adresa webové stránky, díky které můžeme v prohlížeči vyhledat požadovanou webovou stránku. Doména má 3 úrovně. Doména prvního, nebo taky nejvyššího řádu (Top Level Domain - TLD) je doménou uvedenou na konci doménového jména. Domény nejvyššího řádu mají registrované všechny státy (“.cz”, “.sk”), ale jsou i domény nejvyššího řádu generické (“.org”, “.com”). Druhý řád domény je vlastně ta nejdůležitější část celého doménového jména, je to pomyslný název webové stránky (“.spst.”, “.google.”). Třetímu řádu domény se říká subdoména a je to takové rozšíření druhého řádu (“www.“, “org.“). [16]

## Frontend

Frontend je jedno z odvětví tvorby webových stránek a aplikací. Primárně se zaměřuje na tvorbu interaktivních prvků, které uživatelé vidí, a se kterými přímo komunikují na webových stránkách nebo ve webových aplikacích. Hlavním cílem je vytvořit uživatelsky přívětivé prostředí (UI) a prostředí takové, které bude pro uživatele příjemnou zkušeností (UX).

### UX

UX (User experience) se skládá ze **všech interakcí**, které uživatel s produktem nebo službou má. Je to osobní, interní zkušenost, kterou zákazníci procházejí při používání rozhraní produktu.[1]

### UI

UI (User interface) má za úkol vytvořit takové rozhraní, aby uživateli umožnil mít snadnou a příjemnou interakci s webovou stránkou nebo webovou aplikací.

### Frontend technologie

Využívá mnoho programovacích jazyků, k vyzdvižení jsou 3 nejpoužívanější z nich. Těmi jsou HTML, CSS a JavaScript. Tyto jazyky využívá velká většina moderních webových stránek a webových aplikací.

### HTML

HTML (HyperText Markup Language), je jazyk značkovací, který vytvořil Tim Berners-Lee v roce 1991[9]. Jeho hlavním úkolem je tvoření hlavní obsahové kostry všech webových stránek a webových aplikací. Jeho důležitou částí jsou tagy, které slouží k obalování textu a určování jeho důležitosti.

### CSS

CSS (Cascading Style Sheets) je stylovací jazyk, který vytvořil v roce 1994 norský počítačový vědec Håkon Wium Lie. CSS bylo standardizováno až v roce 1996 organizací W3C (World Wide Web Consortium), ve které byl zaměstnán. [10] Hlavní využití CSS je při stylování HTML dokumentů. Stylováním lze definovat rozložení stránky, barvy a třeba i písmo. Lze pomocí něj taktéž tvořit jednoduché animace. Skvělá vlastnost CSS je ta, že umožňuje oddělování kostry stránky a designu, čímž přispívá k lepší uspořádanosti kódu.

### JavaScript

JavaScript je skriptovací jazyk, který roku 1995 vytvořil americký programátor Brendan Eich[11]. JavaScript umožňuje vytvářet dynamicky se aktualizující obsah, ovládat multimédia a animovat obrázky [2]. Má mnoho frameworků a knihoven, které usnadňují tvorbu webových stránek a aplikací.

### JavaScript Framework

JavaScript framework je kolekce knihoven kódů JavaScriptu, které poskytují vývojáři webových aplikací předem napsaný kód pro běžné programovací úlohy. Jsou to struktury s určitým kontextem a pomáhají vytvářet webové aplikace. [3] Nejznámějšími frontend JavaScript frameworky jsou React.js, AngularJS a Vue.js.

## Backend

Backend je „zadní“ část webové stránky nebo webové aplikace. Hlavní rozdíl oproti frontendu je ten, že běží na serveru a zajišťuje logiku, ukládání a zpracování dat do klientské části webu nebo aplikace. Úlohou, kterou backend zajišťuje je obsluhovat vstupy klienta, pracovat s databázemi anebo spravovat operace mezi službami a systémem.

### Technologie backendu

Jelikož je backend tvořen na části serveru, je zde mnoho využívaných technologií. Hlavní je samozřejmě server, na kterém celý backend běží. Backend pracuje i s API (Application Programming Interface) nebo databázemi. Backend lze programovat v mnoho různých programovacích jazycích.

### Server

Stará se o serverovou část webových stránek a webových aplikací. Spravuje data a logiku stránky, která uživatel nevidí. Server interaguje s databázemi za účelem ukládání a načítání dat podle požadavků frontendu. Provádí základní funkce aplikace, zpracovává vstupy z frontendu a určuje, jaká data se mají odeslat zpět. Poskytuje rozhraní API, se kterými může frontend komunikovat. Spravuje ověřování a autorizaci uživatelů a zajišťuje, aby uživatelé měli správná oprávnění k přístupu k určitým datům nebo funkcím.[4]

### API

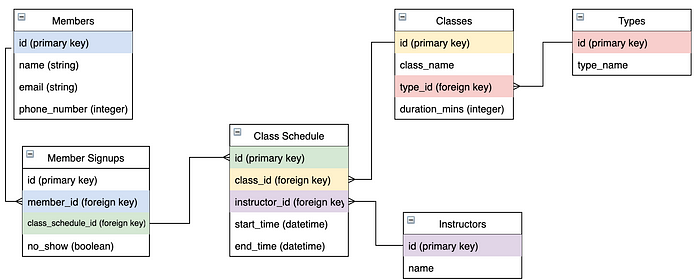
API je rozhraní sloužící pro komunikaci a sdílení dat nebo funkcí. Je to propojení mezi dvěma systémy, kde jedna aplikace přistupuje k datům nebo funkcím druhé aplikace.

### Databáze

Databáze je systém, kde se uschovávají data a lze s nimi manipulovat. Databáze jsou tvořené ve formátu. Sloupce jsou definovány datovými typy a dalšími vlastnostmi. Na druhé straně řádky, jsou již zapsané záznamy. Databáze mají i mnoho modelů a nejznámějšími z nich jsou relační a NoSQL (Not Only Structured Query Language).

### Databáze relační

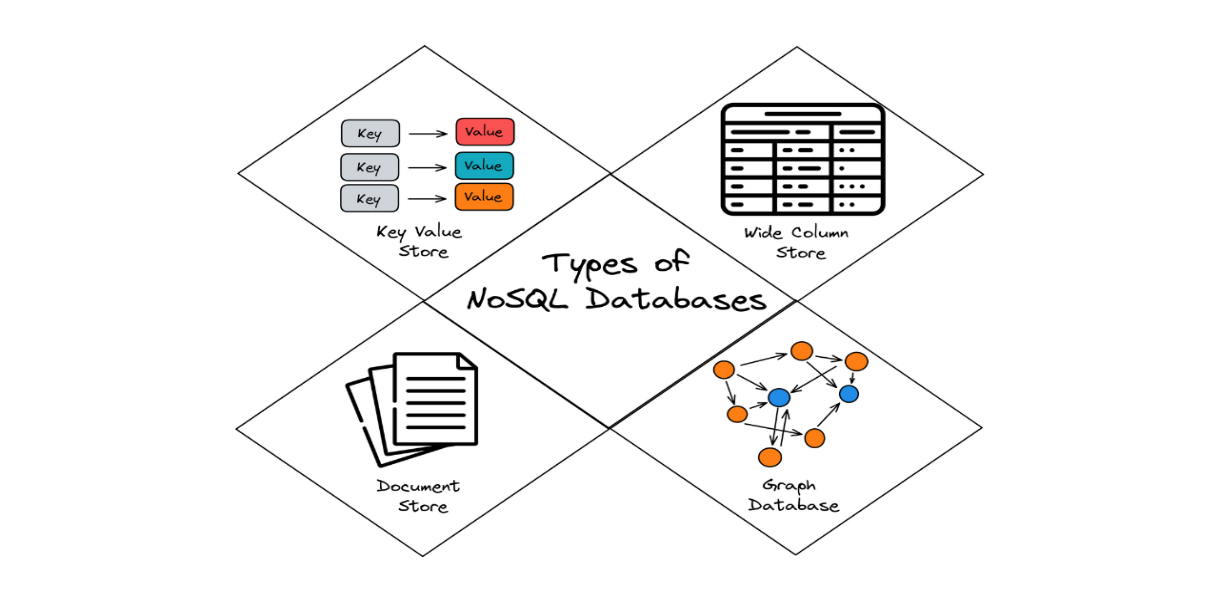
Relační databáze je [databáze](https://wikisofia.cz/wiki/Datab%C3%A1ze) obsahující relace (vztah mezi několika prvky jedné nebo více množin). Je tvořena několika samostatnými tabulkami a vazbami mezi nimi. Každá tabulka obsahuje již strukturovaná [data](https://wikisofia.cz/wiki/Data) o jednom jediném tématu. Relační databáze je založena na [relačním modelu](https://wikisofia.cz/wiki/Popis_prvk%C5%AF_konceptu%C3%A1ln%C3%ADch_model%C5%AF,_smysl_konceptu%C3%A1ln%C3%ADho_modelu_a_p%C5%99%C3%ADklad_modelu_Knihovny). [5]



Obrázek 1.1: Příklad relační databáze [12]

### NoSQL databáze

NoSQL databáze umožňují tvoření databází náročnějších na velké objemy dat, zajišťují škálování, flexibilitu a jsou snadno dostupné. Využívá několik různých formátů ukládání dat (dokumenty, klíč-hodnota, sloupce, grafy).



Obrázek 1.2: Typy NoSQL databází [13]

### MongoDB

**Je databáze s otevřeným zdrojovým kódem orientovaná na dokumenty, která je určena k ukládání velkého množství dat a zároveň umožňuje s těmito daty velmi efektivně pracovat.** Řadí se do kategorie databází NoSQL.[6]

### PHP

PHP (Hypertext Preprocessor) je skriptovací jazyk, který vytvořil dánsko-kanadský programátor Rasmus Lerdorf. Úkoly PHP se provádějí na straně serveru, na kterém je tento jazyk závislý. Fungování PHP má oproti JavaScriptu nevýhodu, protože k provedení každé své nové události musí být vždy prohlížečem znovu odeslán požadavek na server. [7]

### Node.js

[Node.js](https://www.geeksforgeeks.org/nodejs/) je open-source multiplatformní běhové prostředí [JavaScriptu](https://www.geeksforgeeks.org/javascript/" \t "_blank), které umožňuje vývojářům spouštět kód JavaScript na straně serveru. Node.js, který v roce 2009 vytvořil Ryan Dahl, způsobil revoluci v [programování na straně serveru](https://www.geeksforgeeks.org/server-side-client-side-programming/) tím, že nabízí efektivní, událostmi řízený a neblokující model I/O (Input/Output).[8]

## Frontend VS Backend

Hlavní rozdíl mezi fungováním frontendu a backendu je ten, že frontend je spouštěn na obrazovce uživatele a tvoří viditelnou stranu webové stránky, zatímco backend se spouští na serveru a normální uživatel se k němu nemá jak dostat.

|  |  |  |
| --- | --- | --- |
| Aspekt | Frontend | Backend |
| Kde se spouští? | Zařízení uživatele | Server |
| Jaký je cíl? | Vytvoření vizualizace webové stránky a interakce s uživatelem | Zpracovávání dat, logiky a správa funkcí serveru |
| Jaké jsou technologie? | HTML, CSS, JavaScript | PHP, Python, Node.js |
| Jak komunikují? | Pomocí GUI, při odeslání metod POST a GET backendu | S databázemi, při předání API od frontendu |

Tabulka 1.1: Rozdíl mezi frontendem a backendem [14]

# Praktická část

## Výběr projektu

Při výběru projektu jsem se hlavně zaměřil na odvětví, ve kterém se zadání ročníkové práce nachází. Tento projekt je zaměřen na tvorbu webové aplikace, což je mi v mém oboru nejbližší. Zákazníkem byla větší chovatelská stanice ve Zbožíčku ve středočeském kraji. Velkým faktorem bylo tedy také to, že sám vlastním mainskou mývalí kočku, na kterou se chovatelská stanice specializuje. Téma webové stránky jsem si také vybral proto, protože je v tomto odvětví mnoho způsobů a technologií, jak webovou stránku vytvořit.

## Výběr programovacích jazyků

Výběr programovacích jazyků pro tvorbu webové stránky nebyl příliš složitý. Šel jsem nejjednodušší cestou a vybral si základní programovací jazyky jak pro frontend, tak pro backend. Pro tvorbu přední části webové stránky, jsem použil nejpoužívanější značkovací jazyk, HTML. Na stylování HTML jsem použil nejrozšířenější stylovací jazyk, CSS. Ve webové stránce jsou i naimplementované scripty pomocí skriptovacího jazyku, JavaScriptu. Pro tvorbu serverové strany webu jsem využíval skriptovací jazyk, PHP. PHP je využíváno v 80% souborů celé webové stránky. Při tvorbě PHP souborů jsem používal kombinaci s HTML prvky, jelikož většina podstránek na webové stránce používá data z databází, které si za pomoci samotného HTML nemohu brát.

## Výběr vývojových nástrojů

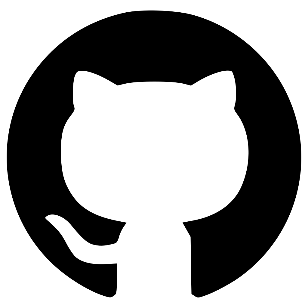
### Výběr editoru kódu

Obsah obrázku symbol, Grafika, Elektricky modrá, řada/pruh

Obsah vygenerovaný umělou inteligencí může být nesprávný.Nástrojů jsem při realizaci ročníkového projektu používal několik. Jakožto editor mých kódů jsem využil Visual Studio Code, to je pravděpodobně nejrozšířenější editor kódů, se kterým pracuje téměř každý tvůrce webových stránek. Visual Studio Code jsem si vybral z jednoduchého důvodu, jelikož je využívaný při školní výuce, tak mám s ním největší zkušenosti a nemusím se učit s nějakým novým, méně používaným editorem.

Obrázek 2.1: Logo Visual Studio Code [17]

### Výběr verzovacího systému

Verzovací systémy se využívají hlavně k lehké správě kódu. Pro můj ročníkový projekt jsem si vybral platformu Github. Při tvorbě webových stránek, ale i všelijakých jiných programů a aplikací je Github téměř povinností. Při práci s Githubem je práce zpříjemněná díky tomu, že tato platforma umožňuje sledování. Github využívá práci s commity. Commity představují vlastně jednotlivé změny ve složkách, souborech a ve všem, co je nahrané na Githubu, což zjednodušuje to, že při tvorbě commitu si tvůrce zadá název a popis jednotlivého commitu a v historii commitů si může dohledat veškeré změny, které kdy udělal. Další výhodou používání platformy Github je přehlednost. Uživatel si může založit několik repozitářů, ale vždycky najde to co potřebuje, protože je vše krásně přehledné a jasné.

Obrázek 2.2: Logo Githubu[18]

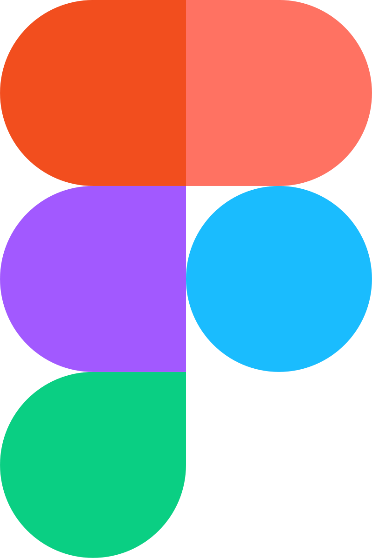
Obsah obrázku text, snímek obrazovky, software

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.3: Příklad změn kódu v Githubu

### Výběr nástroje pro design

Pro vytváření designových prvků, jsem si vybral GIMP a Figmu. GIMP je editor pro rastrové obrázky a vybral jsem si ho hlavně kvůli tomu, že je zdarma. Figma je nástroj, který se používá pro navrhování rozhraní. GIMP i Figmu jsem však mockrát nevyužil a používal jsem je jen na úplné začátku tvorby projektu.



Obrázek 2.4: Logo Figmy [19]

Obrázek 2.5: Logo GIMPu [20]

### Výběr lokálního serveru

Při výběru lokálního serveru jsem se také moc nezapotil. Volba byla jasná, XAMPP. XAMPP je bezplatný open-source software, který napomáhá tvůrcům webových stránek, díky tomu, že přes něj lze jednoduše stránky vytvářet a testovat. Výhoda XAMPPU je ta, že vše funguje pouze lokálně a není zdlouhavě nahrané nikde na hostingu. Pro vývoj stránek v PHP je takový lokální server povinností, jelikož je zapotřebí při tvorbě s PHP. Lokální server se používá při tvorbě dynamických stránek, což XAMPP naprosto splňuje.

Obsah obrázku text, snímek obrazovky, software, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.6: Rozhraní XAMPPU

### Výběr editoru databází

V projektu pracuji s mnoha databázemi. Databáze využívám typu MySQL a ukládám do nich uživatele, novinky nebo třeba i kočky. Pro práci s databázemi jsem se rozhodnul pokračovat s prací s již zmíněným XAMPPEM. Součástí XAMPPU je jednoduché webové rozhraní phpMyAdmin. V phpMyAdmin je možné si vytvořit několik databází nebo tabulek. Velké plus při využívání phpMyAdmin je to, že není potřeba jen psaní SQL příkazů pro tvorbu tabulek nebo vkládání dat do nich. Tvůrce si tedy může jednoduše vše naklikat v grafickém rozhraní a nemusí psát zdlouhavé SQL příkazy.

Obsah obrázku text, software, Počítačová ikona, Webová stránka

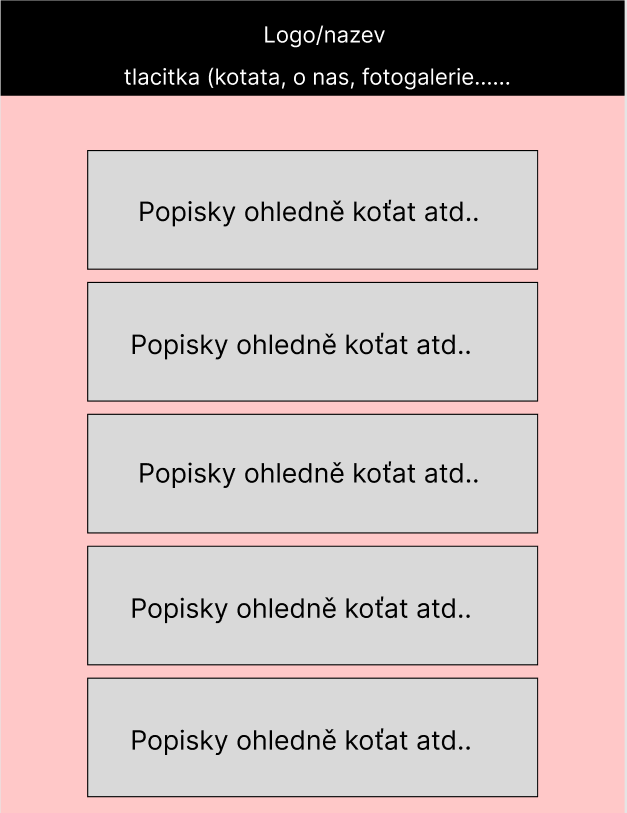
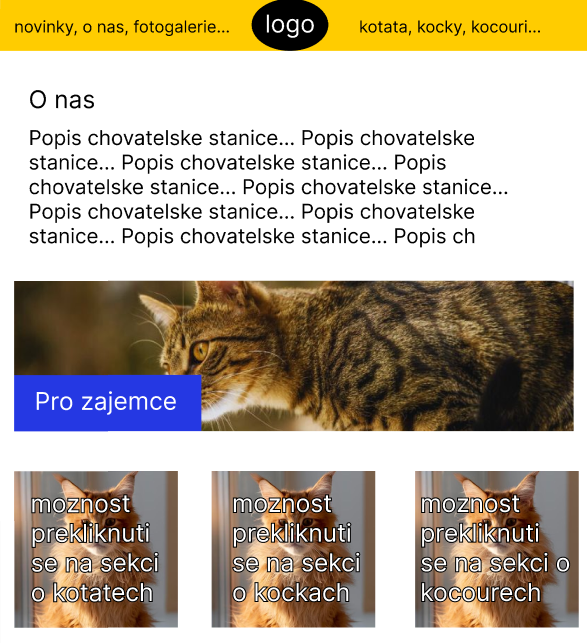
Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.7: Rozhraní phpMyAdmin

## Tvorba webové stránky

### Návrh

Když jsem začal tvořit mojí webovou stránku, tak jsem se začal návrhem hlavní stránky. Pro tvorbu tohoto návrhu jsem použil aplikaci Figma. Ve Figmě to byla moje premiéra, takže jsem se ze začátku lehce trápil, ale po pár naučných videích jsem se na samotný navrh vrhnul. Návrhy jsem vytvořil 2. První návrh byl ve stylu staré stránky mého klienta, tento návrh jsem vytvořil pouze kvůli tomu, kdyby zákazníkovi nevyhovovali velké změny. Při tvorbě druhého návrhu jsem se snažil zahrnout více moderních a stylových prvků. Jelikož zákazník na webové stránce aktuálně nemá žádnou úvodní stránku, chtěl jsem vytvořit něco, co bude pro uživatele jednoduché s možností se jednoduše dostat k tomu, co na stránce opravdu hledá a to jsou samotné kočky. Vytvořil jsem tedy návrh, kde jsou 4 sekce, sekce pro zájemce a 3 sekce s cestou ke kočkám, kocourům a kastrátům, které tato chovatelská stanice vlastní.



Obrázek 2.8: První návrh

Obrázek 2.9: Mnou vytvořený návrh

### 

Obrázek 2.10: Aktuální vzhled úvodní stránky starého webu

### Tvorba navigačního panelu (navbaru)

Tvorbu projektu jsem začal v normálním pořadí, prvně navbar, potom zbytek. Při tvorbě samotného navbaru jsem se snažil držet mého návrhu. V navbaru jsem zanechal téměř stejné odkazy na stránky, jako na starém webu. Byly zde zanechány odkazy na stránky s kočkami, kocoury a kastráty, dále si člověk může vybrat stránku s koťaty a jejich odchovy, novinky chovatelské stanice a kontakt na paní majitelku. V navbaru se může uživatel dostat i k formuláři pro přihlášení a registraci a k přepínači jazyku, ve kterém chce uživatel webovou stránku mít. Navbar je samozřejmě náročný na šířku zařízení, a tak je zde přidáno pro mobilní zařízení takzvané hamburger menu, navigační panel pak vypadá tedy tak, že je zde pouze tlačítko pro přihlášení i s měničem jazyku a samotné hamburger menu, to funguje tak, že když uživatel na tlačítko hamburgeru klikne, tak mu vyjede menu se stejnými linky, jaké by měl na monitoru počítače.

Obsah obrázku text, růžová, snímek obrazovky, Purpurová

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.11: Navbar pro počítače

**Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.**

Obrázek 2.12: Otevřené hamburger menu

Obrázek 2.13: Navbar pro mobilní zařízení

**Obsah obrázku text, růžová, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.**

Obsah obrázku text, Písmo, snímek obrazovky, řada/pruh

Obsah vygenerovaný umělou inteligencí může být nesprávný.Obsah obrázku text, snímek obrazovky, displej, software

Obsah vygenerovaný umělou inteligencí může být nesprávný.Navbar jsem se rozhodl udělat do jednoho souboru a pak ho dynamicky nahrávám do každé jedné stránky. Výhoda tohoto využití je taková, že ušetřím stovky řádků kódu jenom díky tomu, že pomocí jednoho příkazu si navbar na stránku nahraju.

Obrázek 2.14: Dynamické nahrávání navbaru

Obrázek 2.15: Soubor navbar.php

### Tvorba loga

Když jsem měl již vytvořený navbar, něco mi tam chybělo, ano, bylo to logo. Pro tvorbu loga jsem použil program GIMP, s tímto programem jsem se setkal poprvé, a tak jsem tedy nemohl vytvářet žádné velké věci. Vytvořil jsem si tedy jednoduché logo v podobě hlavy oranžové kočky, aby to bylo autentické se známým pohádkovým kocourem, Garfieldem. Samotnou hlavu kočky jsou použil jako faviconu na mé webové stránce a druhou verzi s názvem chovatelské stanice jsem umístil na začátek navbaru, pomocí kterého se může z jakékoliv stránky uživatel přesměrovat zpět na hlavní stránku.





Obrázek 2.16: Logo samotné hlavy (favicon)

Obrázek 2.17: Logo s textem použité v navbaru

### Tvorba formuláře pro přihlášení a registraci

Po zhotovění celého navbaru, jsem se vrhnul rovnou na formulář, kde se může uživatel přihlásit, popřípadě zaregistrovat. Formulář se dá úplně jednoduše otevřít pomocí kliknutí na tlačítko „Přihlásit se“. Při registraci se zadávají 2 věci, uživatelské jméno a heslo. Oboje je ošetřeno na minimální délku, u uživatelského jména jsou to 3 znaky, obohacené ještě o podmínku používaní jen teček a podtržítek v kontextu speciálních znaků, a u hesla to je 6 znaků bez jakéhokoliv omezení. Veškerý uživatelé se ukládají do databáze „users“. V tabulce uživatelů je uložené id každého uživatele (to je generováno automaticky), uživatelské jméno a heslo, které je zahashované a tedy kompletně bezpečné.

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.18: Tabulka "users" s testovacími daty

Obrázek 2.19: Formulář pro registraci

Obrázek 2.20: Formulář pro přihlášení

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.21: Formulář pro registraci

Obrázek 2.22: Formulář pro přihlášení

Situace s formulářem pro přihlášení je úplně stejná jako s navbarem. Celý formulář je udělán do jednoho souboru a je načítán do stránek dynamicky. Díky dynamickému načítání jsem tedy ušetřil další stovky řádků a minuty dlouhého kopírování do každé jednotlivé stránky.

**Obsah obrázku text, snímek obrazovky

Obsah vygenerovaný umělou inteligencí může být nesprávný.**

**¨**

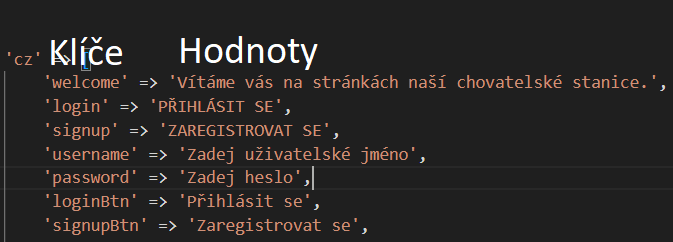
Obrázek 2.23: Soubor logForm.php

****

Obrázek 2.24: Dynamické načítání přihlašovacího formuláře

### Tvorba měniče jazyku

Jedním z úkolů v zadání ročníkového projektu je lokalizace pro minimálně 2 jazyky, češtinu a angličtinu. Hlavním jazykem celé webové stránky je čeština, při přepnutí jazyku na angličtinu se však bez problému celá stránka přeloží a zahraniční uživatelé si bez problému můžou přečíst stejné informace, akorát v angličtině. Při přepínání jazyku využívám uložení jazyku do session a podle zvoleného jazyka se načítají jednotlivé prvky na stránce. Pro překlad jsem využil takový způsob, že si pro každý pevný text na stránce udělám 2 texty do asociativního pole, které funguje na bázi klíčů, které fungují jako identifikátory. Pro vkládání textů jsem používal funkci translate(), která překládala jednotlivé klíče.

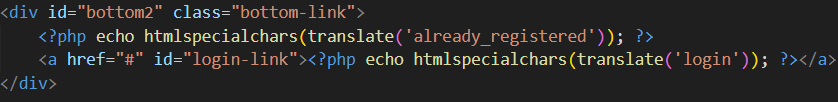


Obrázek 2.25: Asociativní pole pro české texty

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

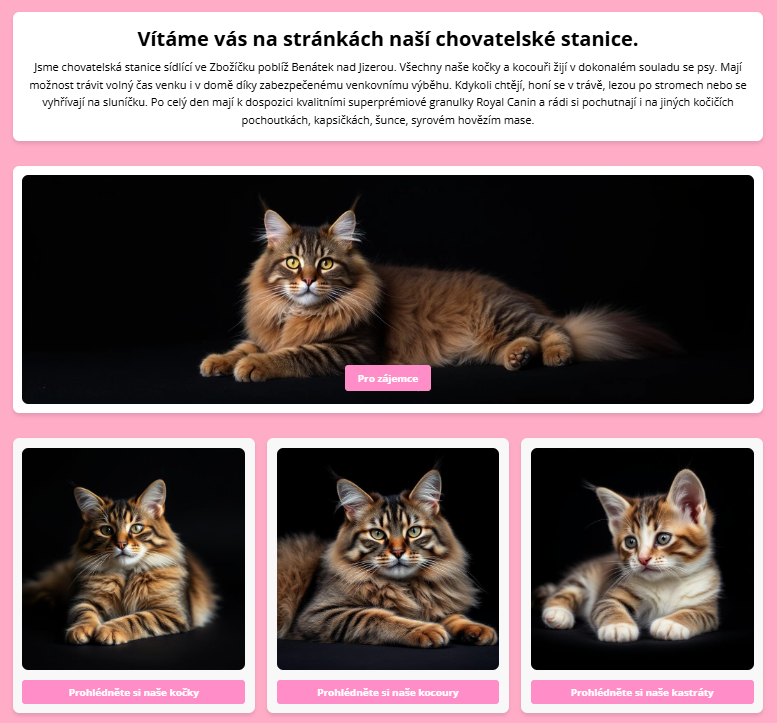
Obrázek 2.26: Asociativní pole pro anglické texty



Obrázek 2.27: Vkládání textů pomocí klíče

### Tvorba úvodní stránky

Při tvorbě hlavní stránky jsem se držel úplně přesně mého návrhu. Stránka je jednoduchá a přehledná a zahrnuje pouze důležité věci. Na úplném začátku stránky je přivítání návštěvníka a stručný popis ohledně prostorů pro kočky, jejich výživy a jejich výchovy. Pod základními informacemi je sekce, kde se může zájemce dozvědět jak probíhá celá adopce, co je k adopci potřeba a taktéž co zájemce po adopci obdrží. Pod sekcí pro zájemce je sekce s odkazy na stránky s všemi kočkami, kocoury a kastráty chovatelské stanice.



Obrázek 2.28: Vzhled úvodní stránky

## Přechod do Reactu

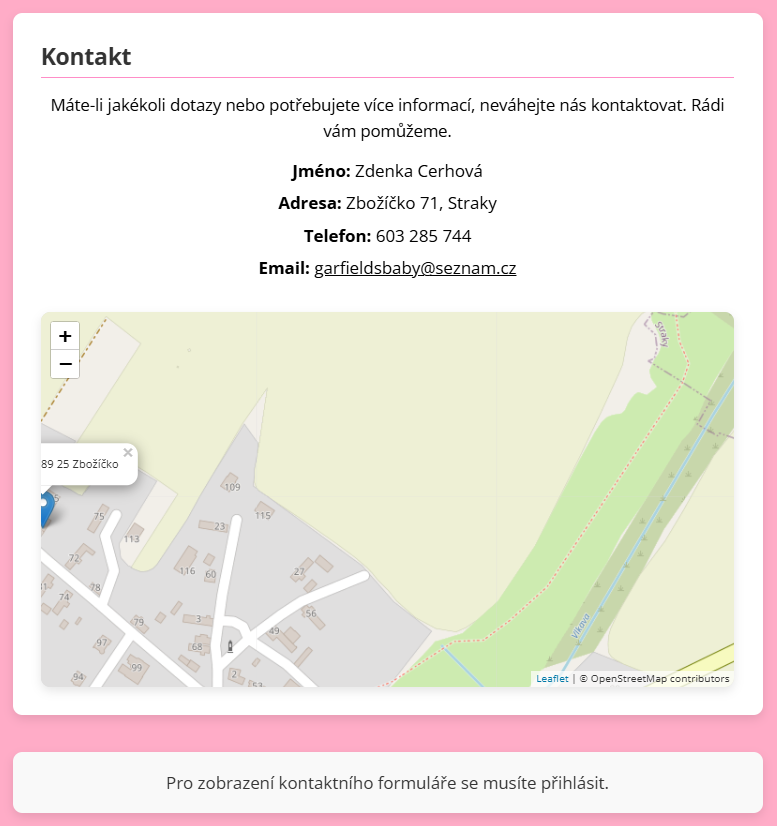
Po vytvoření úvodní stránky nastala dlouhá pauza. Objevila se u mě obrovská prokrastinace a nechuť pokračovat. Když jsem se však opět k tvorbě odhodlal, napadl mě nápad, jak si realizaci projektu zpestřit. Tím zpestřením byl nápad celou webovou stránku dělat ve frameworku React. React je velmi využívaný JavaScriptový framework, který je ideální pro tvorbu webových aplikací. Velký problém však byl v tom, že jsem se v Reactu nedokázal orientovat. Prošel jsem si několik online kurzů, experimentoval jsem při tvorbě webové stránky, ale bylo to až příliš náročné. Po prohraném boji s Reactem jsem se opět vrátil k PHP, ve kterém jsem i dokončil celý projekt.

## Odmítnutí klienta

Se zákazníkem jsme měli již od začátku problém s komunikací. Zákazník pozdě odpovídal na emaily, já se dlouho neozval kvůli prokrastinaci a celkově tedy naše domluva byla na bodu mrazu. Jednoho dne se však zákazník ozval a sdělil mi, že již nemá zájem o mojí webovou stránku. Zákazník si uvědomil, že je na jeho zastaralé webové stránky zvyklí a nechce podstoupit změnu. Pro někoho by toto znamenal konec, já jsem však díky tomuto chytil nový impuls a začal jsem pilně webové stránky tvořit, abych dokázal, že mi na mém projektu opravdu záleží.

## Tvorba kontaktní stránky

Nabitý motivací jsem se ihned pustil do tvorby teď již mé webové stránky. Jako dalším prvkem na stránce jsem vytvořil stránku, kde si uživatel může dohledat veškeré údaje pro případné zkontaktování majitele. Na stránce se nachází hlavní údaje, které by zákazník potřeboval pro kontakt a to je: jméno majitele, adresa chovatelské stanice, telefonní číslo a emailová adresa. Jedním úkolem ze zadání ročníkové práce je ten, že musí být na webové stránce interaktivní mapa. Interaktivní mapu jsem přidal díky open-source JavaScript knihovně, Leaflet JS. Tato interaktivní mapa ukazuje přímou polohu chovatelské stanice a lze s ní posouvat nebo třeba oddalovat.



Obrázek 2.29: Vzhled kontaktní stránky



Obrázek 2.30: Nahrávání knihovny Leaflet JS

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.31: Inicializace interaktivní mapy

Mapou kontaktní stránka však nekončí, pod interaktivní mapou je umístěn kontaktní formulář. Tento kontaktní formulář je člověku skryt, dokud není na webové stránce přihlášen. Veškeré dotazy zaslané z kontaktního formuláře jsou uloženy do databáze. Při reálném využívání webové stránky, bych zaslané dotazy přesměrovával do emailu majitele, místo databáze.

Obsah obrázku text, snímek obrazovky, číslo, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.32: Kontaktní formulář

Obsah obrázku text, Písmo, číslo, řada/pruh

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.33: Tabulka "contacts" s testovacími daty

Obrázek 2.34: Vzhled kontakt stránky

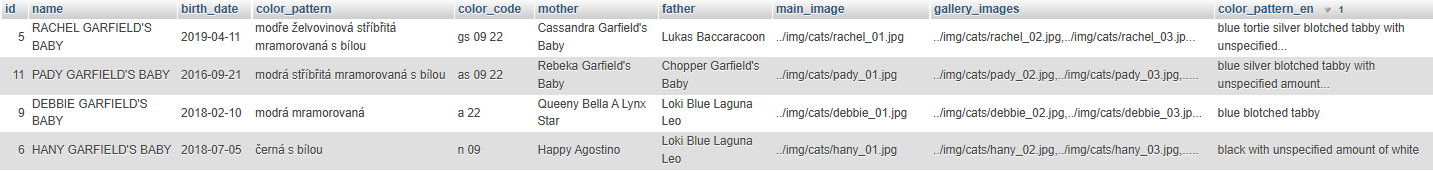
## Tvorba stránky o kočkách

Obsah obrázku text, pes, savec

Obsah vygenerovaný umělou inteligencí může být nesprávný.Na začátku tvorby webové stránky, kde jsou ukázány všechny kočky chovatelské stanice nastal opět další problém, tentokrát v závislosti na odmítnutí klienta. Jelikož jsem ztratil zájem klienta, neměl jsem žádná data ze staré webové stránky. Fotky koček jsou však velice důležité a musel jsem si tedy poradit. Abych získal veškerá data všech koček, musel jsem si tedy všechny fotografie koček stáhnout jednotlivě ze staré webové stránky. Po získání všech potřebných dat, jsem se rozhodl na stránce tvořit pro každou kočku svůj vlastní blok, kde je jeden hlavní obrázek a 4 další pohledy na kočku. V bloku kočky jsou vepsány všechny důležité informace o kočce a to: jméno kočky, datum narození, barva, EMS kód (tyto kódy vyjadřují zbarvení, vzor a pigment bílé barvy u srsti koček) a dále je zde i rodokmen každé kočky (tím je myšlena matka a otec kočky). Tento design bloků se mi líbí v tom, že je celá stránka přehledná a každá kočka má svůj blok, který není ničím překrýván nebo není s ničím spojen.

Obrázek 2.35: Vzhled stránky "Kočky"

Všechny kočky jsou samozřejmě uloženy i do databáze. Díky datům v databázi jsem měl možnost načítat kočky dynamicky, a ne jednotlivě a zdlouhavě jednu po jedné do HTML kódu. Díky dynamickému načítání dat z databáze, jsem uložil další stovky řádků kódu a celkově jsem i zjednodušil celkovou práci se stránkou koček.

****

Obrázek 2.36: Tabulka "cats" s daty koček

**Obsah obrázku text, snímek obrazovky, software

Obsah vygenerovaný umělou inteligencí může být nesprávný.**

Obrázek 2.37: Dynamické vkládání koček do bloků

## Tvorba stránky o kocourech

Stránka s kocoury byl stejný příběh, veškeré obrázky kocourů jsem si musel ručně stáhnout ze staré webové stránky a následně udělat stejný postup jako u stránky s kočkami. Všechna data jsem vložil do databáze a následně je vložil stejným způsobem do stejných bloků jako na stránce o kočkách.

## Tvorba stránky o kastrátech

Jelikož je chovatelská stanice aktivní i na soutěžích krásy, kastráti chovatelské stanice nasbírali už několik oceněních. Do bloků pro kastráty jsou tedy přidány i výstavy, na kterých byli kastráti přítomní, a jejich úspěchy na samotných výstavách.

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.38: Příklad výpisu informací o kastrátu i s výstavami

Všechny obrázky na stránkách o kočkách, kocourech nebo kastrátech si může uživatel jednoduše zvětšit pomocí pouhého kliknutí. Při situaci, kde je obrázek již zvětšený, lze si obrázek libovolně přibližovat a posouvat, jak jen uživatel chce.

Obsah obrázku text, snímek obrazovky, Malé a středně velké kočky, Kočkovití

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.39: Zvětšený obrázek

## Tvorba stránky pro změnu profilu

Jelikož je na stránce možnost přihlašování, musí být zde i jiné využití. Rozhodl jsem se tedy přidat formulář, kde si může uživatel změnit uživatelské jméno a heslo. Ve formuláři jsou vstupy ošetřeny tak, aby nemohl uživatel zadat stejné heslo, nemohl zadat uživatelské jméno kratší než 3 znaky, nebo třeba nemohl změnit heslo, dokud správně nenapíše heslo aktuální. Pro výpis chyb jsem zvolil přidání alertů, které po 5 sekundách zmizí. Po změně ať už uživatelského jména nebo hesla se aktualizované informace ihned přepíší v databázi uživatelů.

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.40: Formulář pro změnu uživatelského jména a hesla s příkladným výpisem chyby

## Tvorba stránky pro novinky

Podstránka o novinkách je první věc, kterou uživatel spatří, hned po příchodu, na staré webové stránce. Já se však rozhodl jít jinou cestou a novinky jsem udělal ve formě podstránky. U novinek jsem postupoval stejně jako u obrázků v podstránkách o kočkách, kocourech nebo kastrátech. Veškeré články jsem postupně nahrával do databáze novinek. Při tvorbě ročníkového projektu jsem se snažil o jednu důležitou věc, a to o tu, aby byl veškerý kód jednoduchý, časově nenářočný a pro správce webové stránky přívětivý. Proto jsou tedy články opět nahrávány dynamicky a pro ještě větší usnadnění práce jsou články řazeny podle datumu nahrání. V téměř každé databázi jsou data řazeny podle id, tady však ne. Hlavním důvodem je to, že kdyby správce webové stránky chtěl nahrát dodatečně nějaký článek, nemusí přeskládávat celou databázi, aby byly články ve správném pořadí.

Obsah obrázku text, Písmo, číslo, řada/pruh

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.41: Vzhled stránky s články

Obrázek 2.42: Databáze "news" s články

Obsah obrázku text, snímek obrazovky

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obrázek 2.43: Dynamické vkládání článků

## Tvorba stránky s odchovy

Stránka s odchovy má jednu podstatnou věc, a to takovou, že je potřeba zřetelně vidět, jaký vrh chce uživatel otevřít a jelikož je to webová stránka pro velkou chovatelskou stanici, tak těch vrhů bylo několik řad. Proto jsem se tedy hlavně zaměřil na jednoduchý, zároveň ale dost jasný styl stránky, aby bylo vždy jasné, jaký vrh si chci rozkliknout.

Obrázek 2.44: Vzhled stránky s odchovy

Závěr

Vytvořená šablona maturitních prací obsahuje formální požadavky maturitních prací na SPŠT Třebíč. Jedná se zejména o upravené styly v dokumentu, podrobný popis jednotlivých částí maturitní práce a jejího obsahu, snadno editovatelné záhlaví a zápatí s automatickým číslováním stránek a propojení stylů se seznamy a obsahem.

Seznam zkratek

UI – User interface

UX – User experience

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

PHP – Hypertext Preprocessor

GUI – Graphical user interface

SQL – Structured Query Language

NoSQL – Not Only Structured Query Language

API – Application Programming Interface

I/O – Input/Output

Seznam použitých symbolů a zkratek

Seznam použitých zdrojů

1. ZAPPA, Letizia. *What is User Experience? Overview and example.* Online. 2019. Dostupné z: <https://userreport.com/blog/user-experience/> [cit. 2024-11-13].
2. MDN WEB DOCS. *What is JavaScript?,* MDN Web Docs. Online. 2024. Dostupné z: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript> [cit. 2024-11-13].
3. ASSEMBLY, General. *What are JavaScript Frameworks? Overview and example*. Online. 2024. Dostupné z: <https://generalassemb.ly/blog/what-is-a-javascript-framework/> [cit. 2024-11-13].
4. QUORA. *What is a front-end server and back-end server?* Online. Dostupné z: <https://www.quora.com/What-is-a-front-end-server-and-back-end-server?__cf_chl_tk=2guB26BRaaFhUiDQpeDP4vvTyTz2hMMyCRbOKsFqQPc-1731536887-1.0.1.1-O84OTfF5L8UBrt4S6j47a4NWfg96Vp0KHXrBzFWeGEo>. [cit. 2024-11-13].
5. UNIVERZITA KARLOVA*. Relační databáze,* Wikisofia. Online. 2013. Dostupné z: [https://wikisofia.cz/wiki/Rela %C4%8Dn%C3%AD\_datab%C3%A1ze](https://wikisofia.cz/wiki/Rela%09%C4%8Dn%C3%AD_datab%C3%A1ze). [cit. 2024-11-14]
6. GeeksforGeeks**.** *MongoDB Working and Features,* GeeksforGeeks. Online. 2024. Dostupné z: <https://www.geeksforgeeks.org/what-is-mongodb-working-and-features/>. [cit. 2024-11-21].
7. **WEB & MEDIA.** *PHP část I. - Úvod do jazyka,* Builder.cz - Informační server o programování*.* Online. Dostupné z: <https://www.builder.cz/rubriky/php/php-cast-i-uvod-do-jazyka-155594cz>. [cit. 2024-11-14].
8. GEEKSFORGEEKS. *Node.js Introduction,* GeeksforGeeks*.* Online. 2024. Dostupné z: <https://www.geeksforgeeks.org/node-js-introduction/>. [cit. 2024-11-21].
9. CAKAL, Yasin*.* ***The History of HTML,* Code of Code***.* Online. 2022. Dostupné z: <https://codeofcode.org/lessons/the-history-of-html/#google_vignette>. [cit. 2024-12-12].
10. **BOS, Bert.** *A Brief History of CSS,* World Wide Web Consortium (W3C). Online. 2016. Dostupné z: <https://www.w3.org/Style/CSS20/history.html>. [cit. 2024-12-12].
11. MACMANUS, Richard. *1995: The Birth of JavaScript,* Cybercultural. Online. 2020. Dostupné z: <https://cybercultural.com/p/1995-the-birth-of-javascript/>. [cit. 2024-12-12].
12. LOGAN, Claire. *3 Relation Data Model Examples*, Medium. Online. 2021. Dostupné z: <https://medium.com/@claire_logan/3-relational-data-model-examples-c9f70c61588c>. [cit. 2024-12-12].
13. AGARWAL, Umang. *Types of NoSQL databases.* LinkedIn. Online. 2022. Dostupné z: <https://www.linkedin.com/pulse/types-nosql-databases-umang-agarwal>. [cit. 2025-01-23].
14. GEEKFORGEEKS. *Frontend vs Backend*. GeeksforGeeks*.* Online. 2024. Dostupné z: <https://www.geeksforgeeks.org/frontend-vs-backend/>. [cit. 2025-01-23].
15. VEDOS HOSTING. Můj první web. Online. VEDOS HOSTING. Dostupné z: <https://vedos.cz/zaciname/?utm_source=google&utm_medium=cpc&utm_campaign=CS%20%7C%20Hosting%20%7C%20Search&utm_id=1539916944&gad_source=1&gclid=Cj0KCQjwhYS_BhD2ARIsAJTMMQZ7mY4Ui1GCwVTVxRvpzzPlIQYCEf-6JSQXYJNrYnHtSVaFzU8TY-saAs2jEALw_wcB>. [cit. 2025-03-24].
16. VIZUS. Co to je doména a k čemu slouží? Online. 2006. Dostupné z: <https://www.vizus.cz/spoluprace/knowhow/co-to-je-domena-a-k-cemu-slouzi/>. [cit. 2025-03-24].
17. *Visual Studio Code*. Online. In: Wikipedia, Stránka byla naposledy editována 18. března 2025 v 10:01. Dostupné z: <https://en.wikipedia.org/wiki/Visual_Studio_Code>. [cit. 2025-03-25].
18. *Github*. Online. In: Wikipedia, Stránka byla naposledy editována 25. března 2025 v 16:27. Dostupné z: <https://en.wikipedia.org/wiki/GitHub>. [cit. 2025-03-25].
19. *Figma*. Online. In: Wikipedia, Stránka byla naposledy editována 26. února 2025 v 10:47. Dostupné z: <https://en.wikipedia.org/wiki/Figma>. [cit. 2025-03-25].
20. *GIMP*. Online. In: Wikipedia, Stránka byla naposledy editována 24. března 2025 v 11:22. Dostupné z: <https://en.wikipedia.org/wiki/GIMP>. [cit. 2025-03-25].

normou ČSN ISO 690 Bibliografické citace

Seznam obrázků

[Obrázek 1: Příklad relační databáze [12] 12](#_Toc193784122)

[Obrázek 2: Typy NoSQL databází [13] 12](#_Toc193784123)

Seznam tabulek

[Tabulka 1 Rozdíl mezi frontendem a backendem 14](#_Toc183690134)

Seznam příloh

Prázdná šablona maturitní práce

Nad seznam příloh

Sjednotit k textu

Stačí v textu při prvním objevení