

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky



# **Vývoj webových aplikací pomocí umělé intelligence**

**BAKALÁŘSKÁ PRÁCE**

Studijní program: Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Autor: Vít Koutenský

Vedoucí bakalářské práce: Ing. Richard Antonín Novák, Ph.D.

Praha, květen 2025

## **Poděkování**

Chtěl bych poděkovat svému vedoucímu Ing. Richardu Novákovi, Ph.D. za rady a především trpělivost. Dále bych také rád poděkoval své rodině a přátelům za podporu v průběhu psaní práce.

## **Abstrakt**

Bakalářská práce se zabývá užitím umělé inteligence (AI) při vývoji webů a webových aplikací. Je zaměřena především na přínos AI v rámci efektivity a kvality vyvíjených produktů. Jejím cílem je analýza a komparace přístupů využívajících AI s přístupy tradičními. Rozdíly jsou hledány jednak v oblasti vývoje kódu, kde je porovnáván přístup s využitím AI, tradiční přístup bez AI a tvorba pomocí systému pro správu obsahu webů, jednak v oblasti tvorby vizuálního obsahu, kde jsou porovnávány tradiční přístupy s obsahem generovaným pomocí AI. Při porovnání jsou zohledňována kritéria jako kvalita výstupu, technická náročnost, doba vývoje nebo relevance a estetika a další.

Z výsledků bakalářské práce lze vydodit závěr, že AI je při vývoji kódu webu, v rámci limit práce, přínosem, protože je schopno zkrátit dobu vývoje při zachování dostatečné kvality výstupu ve spojení se snížením technické náročnosti procesu. Při tvorbě vizuálního obsahu vychází AI jako varianta podřazená vůči tradičním přístupům, jelikož i přes zachování dostatečné relevance a estetiky vytváří příliš velké množství nedostatků ve kvalitě výstupu a právně-etických aspektech. V rámci diskuse jsou pak rozebrána doporučení pro zavedení AI do vývoje webů a limity práce jako takové.

## **Klíčová slova**

web, vývoj webů, umělá inteligence (AI), generativní AI, generování obsahu, efektivita

## **Abstract**

The bachelor thesis deals with the use of artificial intelligence (AI) in the development of websites and web applications. It focuses mainly on the contribution of AI in the efficiency and quality of the developed products. It aims to analyse and compare AI approaches with traditional approaches. Differences are sought both in code development, where the AI-assisted approach, the traditional non-AI approach and development using a web content management system are compared, and around visual content creation, where traditional approaches are compared with AI-generated content. The comparison considers criteria such as quality of output, technical complexity, development time or relevance and aesthetics, among others.

From the results of the bachelor thesis, it can be concluded that AI is beneficial in web code development, within the limit of the thesis, as it is able to reduce development time while maintaining sufficient output quality in conjunction with reducing the technical difficulty of the process. In visual content development, AI comes out as an inferior option to traditional approaches as it creates too many shortcomings in output quality and legal-ethical aspects despite maintaining sufficient relevance and aesthetics. Recommendations for introducing AI into web development and the limits of the work itself are then discussed.

## **Keywords**

web, web development, artificial intelligence (AI), generative AI, content generation, effectivity

# Obsah

Úvod .....	11
1 Historický vývoj a aktuální stav webových aplikací a umělé inteligence.....	13
1.1 Historie a současné trendy ve vývoji webových aplikací .....	13
1.1.1 Softwarové inženýrství ve vztahu k webovým aplikacím .....	13
1.1.2 Historie vývoje webových aplikací .....	14
1.1.3 Rozdělení vývoje webové aplikace .....	19
1.1.4 Průběh vývoje webové aplikace.....	20
1.1.5 Časté problémy s vývojem webových aplikací.....	23
1.2 Úvod do umělé inteligence a rozsáhlých jazykových modelů .....	23
1.2.1 Umělá inteligence.....	23
1.2.2 Velké jazykové modely.....	25
1.2.3 Technologie umělé inteligence využitelné při vývoji webových aplikací.....	28
2 Praktické zhodnocení přínosů vývoje webových aplikací pomocí umělé inteligence .....	30
2.1 Použité nástroje pro praktickou část.....	30
2.2 Metodika praktické části.....	31
2.2.1 Zvolená kritéria pro komparaci .....	32
2.2.2 Zvolená porovnávací metoda: Saatyho metoda .....	33
2.3 Vývoj kódu .....	34
2.3.1 Vývoj bez asistence AI.....	34
2.3.2 CMS vývoj.....	40
2.3.3 AI asistovaný vývoj.....	42
2.3.4 Souhrn .....	51
2.4 Tvorba vizuálního obsahu.....	54
2.4.1 Obrázky získané bez použití AI.....	55
2.4.2 Obrázky generované pomocí AI .....	57
2.4.3 Souhrn .....	63
2.5 Multikriteriální porovnání přístupů .....	68
2.5.1 Zvolení cílů.....	68
2.5.2 Výpočet vah kritérií .....	68
2.5.3 Porovnání přístupů .....	71
2.5.4 Výsledky .....	74
Závěr .....	76
Diskuse .....	77

Použitá literatura .....	79
Přílohy .....	I
Příloha A: Odkazy na webové stránky .....	I
Příloha B: Odkazy na GitHub repositáře .....	I

## **Seznam obrázků**

Obrázek 1 - Časová osa historie webu (Barning, 2013) .....	19
Obrázek 2 - SiteMap webu Pila Zachotín (autor) .....	34
Obrázek 3 - Ukázka struktury headeru vytvořená bez AI (autor) .....	35
Obrázek 4 - Ukázka CSS stylování sekce hero vytvořené bez AI (autor).....	37
Obrázek 5 - Vzhled domovské stránky vytvořená bez AI (autor) .....	38
Obrázek 6 - JS logika pro navigaci stránky vytvořená bez AI (autor).....	39
Obrázek 7 - JS logika kalkulátoru funkce update vytvořená bez AI (autor) .....	40
Obrázek 8 - Vzhled šablony využívané v nástroji WordPress (autor) .....	41
Obrázek 9 - Vzhled úvodní stránky při vývoji s CMS (autor).....	42
Obrázek 10 - Vzhled headeru při vývoji s AI (autor) .....	43
Obrázek 11 - Vzhled headeru při vývoji bez AI (autor) .....	44
Obrázek 12 - Vzhled headeru při vývoji s CMS (autor) .....	44
Obrázek 13 - Vzhled headeru při vývoji s AI (autor) .....	44
Obrázek 14 - Vzhled headeru při vývoji bez AI (autor) .....	44
Obrázek 15 - Vzhled headeru při vývoji s CMS (autor) .....	44
Obrázek 16 - Vzhled hero při vývoji s AI (autor) .....	45
Obrázek 17 - Vzhled hero při vývoji bez AI (autor) .....	46
Obrázek 18 - Vzhled hero při vývoji s CMS (autor) .....	46
Obrázek 19 – Vzhled sekce ikon při vývoji s AI (autor).....	47
Obrázek 20 – Vzhled sekce ikon při vývoji bez AI (autor) .....	47
Obrázek 21 - Vzhled sekce ikon při vývoji s CMS (autor).....	47
Obrázek 22 - Vzhled hero2 při vývoji s AI (autor).....	48
Obrázek 23 - Vzhled hero2 při vývoji bez AI (autor).....	48
Obrázek 24 - Vzhled hero2 při vývoji s CMS (autor) .....	48
Obrázek 25 - Vzhled kalkulátoru při vývoji s AI (autor) .....	49
Obrázek 26 - Vzhled sekce 1 kalkulátoru při vývoji bez AI (autor) .....	50
Obrázek 27 - Vzhled sekce 2 kalkulátoru při vývoji bez AI (autor) .....	50
Obrázek 28 - JS logika kalkulátoru vytvořeného s AI (autor).....	50
Obrázek 29 - Fotografie expedice řeziva bez AI (autor) .....	55
Obrázek 30 - Fotografie štěpkování bez AI (autor) .....	55
Obrázek 31 - Fotografie expedice řeziva bez AI (autor) .....	55
Obrázek 32 – Fotografie štěpkování bez AI (autor) .....	55
Obrázek 33 - Fotografie dřevěného srdce z fotobanky bez AI (autor) .....	56
Obrázek 34 - Ikona listnatého stromu bez AI (autor) .....	56
Obrázek 35 - Ikona pilového kotouče bez AI (autor) .....	56
Obrázek 36 - Logo firmy bez AI (autor) .....	56
Obrázek 37 - Mapa expedice bez AI (autor) .....	57
Obrázek 38 - Fotografie tým s AI (autor) .....	58
Obrázek 39 - Fotografie tým bez AI (autor) .....	58
Obrázek 40 - Fotografie sušárna s AI (autor).....	58
Obrázek 41 - Fotografie sušárna bez AI (autor) .....	58
Obrázek 42 - Fotografie expedice řeziva s AI (autor).....	59
Obrázek 43 - Fotografie expedice bez AI (autor) .....	59
Obrázek 44 – Fotografie štěpkování s AI (autor) .....	59

Obrázek 45 - Fotografie štěpkování bez AI (autor) .....	59
Obrázek 46 - Fotografie mistr s AI (autor).....	60
Obrázek 47 - Fotografie mistr bez AI (autor) .....	60
Obrázek 48 - Fotografie dřevěného srdce s AI (autor).....	61
Obrázek 49 - Fotografie dřevěného srdce z fotobanky bez AI (autor) .....	61
Obrázek 50 - Ikona pilového kotouče s AI (autor) .....	62
Obrázek 51 - Ikona listnatého stromu s AI (autor).....	62
Obrázek 52 - Ikona pilového kotouče bez AI (autor) .....	62
Obrázek 53 - Ikona listnatého stromu bez AI (autor) .....	62
Obrázek 54 - Logo firmy s AI (autor) .....	62
Obrázek 55 - Logo firmy bez AI (autor).....	62
Obrázek 56 - Mapa expedice s AI (autor) .....	63
Obrázek 57 - Mapa expedice bez AI (autor) .....	63
Obrázek 58 - Výstup z ChatGPT 4.5 (autor) .....	65
Obrázek 59 - Grafické znázornění vah stanovených kritérií pro vývoj kódu (autorské zpracování) .....	69
Obrázek 60 - Grafické znázornění vah stanovených kritérií pro tvorbu vizuálního obsahu (autorské zpracování) .....	70

## **Seznam tabulek**

Tabulka 1 - Technická náročnost vývoje (autorské zpracování).....	51
Tabulka 2 - Časová náročnost vývoje (autorské zpracování) .....	52
Tabulka 3 - Výsledky W3C html validace (autorské zpracování) .....	53
Tabulka 4 - Hodnocení výkonu webové stránky na mobilních zařízeních (autorské zpracování) .....	54
Tabulka 5 - Hodnocení výkonu webové stránky na počítači (autorské zpracování) .....	54
Tabulka 6 - Relevance vizuálního obsahu (autorské zpracování) .....	64
Tabulka 7 - Kvalita rozlišení vizuálního obsahu (autorské zpracování) .....	66
Tabulka 8 - Kvalita velikosti vizuálního obsahu (autorské zpracování) .....	67
Tabulka 9 - Hodnocení kritérií podle Saatyho metody (autorské zpracování) .....	68
Tabulka 10 - Párové srovnání a výpočet vah kritérií pro vývoj kódu (autorské zpracování) .....	68
Tabulka 11 - Párové srovnání a výpočet vah kritérií pro tvorbu vizuálního obsahu (autorské zpracování) .....	69
Tabulka 12 - Hodnocení přístupů k vývoji z hlediska zvolených kritérií před započtením vah (autorské zpracování) .....	72
Tabulka 13 - Hodnocení přístupů k vývoji z hlediska zvolených kritérií po započtení vah (autorské zpracování) .....	72
Tabulka 14 - Hodnocení přístupů k tvorbě vizuálního obsahu z hlediska zvolených kritérií před započtením vah (autorské zpracování) .....	73
Tabulka 15 - Hodnocení přístupů k tvorbě vizuálního obsahu z hlediska zvolených kritérií po započtení vah (autorské zpracování) .....	74

## **Seznam zkrátek**

JS – JavaScript

PHP - Hypertext Preprocessor

CMS – Content Management System

AI – Artificial Intelligence

URL – Uniform Resource Locator

HTML - HyperText Markup Language

HTTP - Hypertext Transfer Protocol

SPA – Single Page Application

CSS – Cascading Style Sheet

DOM – Document Object Model

SEO – Search Engine Optimization

JSON - JavaScript Object Notation

AJAX - Asynchronous JavaScript And XML

PaaS – Platform as a Service

ML – Machine Learning

LLM – Large Language Model

RLHF - Reinforcement Learning from Human Feedback

VSC – Visual Studio Code

# Úvod

Vývoj webů a webových aplikací představuje dynamické propojení technologií, kreativity a inovací, které se neustále vyvíjí v souladu s novými trendy a potřebami uživatelů. Tato bakalářská práce se zaměřuje na zkoumání toho, jak umělá inteligence (AI) ovlivňuje způsob, jímž jsou weby a webové aplikace navrhovány, vyvíjeny a optimalizovány. AI poskytuje pokročilé nástroje vedoucí ke zefektivnění a zrychlení práce vývojářů, ale současně umožňuje vznik personalizovaného obsahu pro weby jako takové.

Struktura práce je rozdělena do dvou částí. Teoretická část se zaměřuje na historický vývoj webových aplikací od jejich počátků až po současnost. Popisuje technologické pokroky, které měly za důsledek rozvoj webu jako takového a analyzuje kroky a strukturu ve vývoji webových aplikací. V této části práce jsou rovněž představeny časté problémy, vznikající při vývoji webových aplikací. Lze zde také nalézt informace o umělé inteligenci a jejích klíčových trendech a nástrojích, které lze použít při vývoji webových aplikací.

Praktická část práce je zaměřena na komparativní experimenty, v jejichž rámci byly porovnány webové aplikace vytvořené třemi různými metodami programování – tradičním, s využitím AI a prostřednictvím systému pro správu obsahu webů. Srovnání se zaměřuje na analýzu rozdílů v oblasti kódu a grafických prvků. Jednotlivé výstupy jsou následně hodnoceny pomocí multikriteriální analýzy, která zkoumá technickou a časovou náročnost, kvalitu výsledného řešení a další relevantní parametry. Cílem této části práce je nejen poukázat na výhody využití umělé inteligence, jako jsou vyšší efektivita práce a časová úspora, ale zároveň identifikovat její současná omezení, především v oblastech kontroly kvality a kreativity.

Závěr práce shrnuje klíčová zjištění vyplývající jak z teoretické části, tak z praktického porovnání přístupů ke tvorbě webových stránek pomocí tradičního programování, nástrojů umělé inteligence a systému pro správu obsahu webů. Na základě multikriteriální analýzy jsou zdůrazněny výhody zavedení AI, především časová efektivita a zjednodušení pracovních postupů, stejně jako její aktuální limity, například v oblasti kvality výstupu a omezené flexibility. Práce zároveň formuluje doporučení pro smysluplné využití AI při vývoji webových aplikací a nastiňuje možné směry dalšího rozvoje v této oblasti.

## Cíl práce

Hlavním cílem této bakalářské práce je analyzovat a porovnat různé přístupy k vývoji webových aplikací se zaměřením na využití umělé inteligence. Práce se soustředí na porovnání kódu vytvářeného tradiční cestou, pomocí AI nástrojů a v prostředí systému pro správu obsahu webů, přičemž hodnotí technickou náročnost, dobu vývoje a kvalitu výstupů. Dále zkoumá přínosy umělé inteligence při generování vizuálního obsahu webu, a posuzuje její dopad na časovou náročnost a výslednou úroveň zpracování. Na základě získaných poznatků pak práce formuluje doporučení pro následující začlenění AI do vývoje webových aplikací.

## **Metody**

V rámci této práce byly aplikovány následující metody vědecké práce:

- Literární rešerše vědeckých článků: Literární rešerše umožní autorovi získat přehled o současných přístupech k vývoji webových aplikací s využitím AI, klasického kódování a systému pro správu obsahu webů. Nabídnou pohled na aktuální trendy, možnosti i omezení jednotlivých vývojových přístupů.
- Metoda komparace: Kódy a obrázky vytvořené pomocí AI jsou porovnány s řešeními manuálními nebo vytvořenými pomocí systému pro správu obsahu webů, přičemž jsou analyzovány jejich silné a slabé stránky.
- Experimentální metoda: Pro oblast vývoje kódu a pro oblast tvorby vizuálního obsahu byl proveden experiment, kdy byl stejný úkol vyřešen za pomoci AI a jejich alternativ. Výsledky byly následně porovnány.
- Saatyho metoda: Použita pro vícekriteriální hodnocení kvality, efektivity a časové náročnosti jednotlivých přístupů.
- Metody analýzy a syntézy: Použito pro získání přehledu o jednotlivých částech tématu a následného spojení vybraných částí v celková zjištění.
- Metody konkretizace a abstrakce: Zvolené párové metody umožnily autorovi vyčlenění podstatných informací jednotlivých zjištění a jejich aplikaci v konkrétních případech pro ověření relevantnosti výsledků.

# 1 Historický vývoj a aktuální stav webových aplikací a umělé inteligence

## 1.1 Historie a současné trendy ve vývoji webových aplikací

### 1.1.1 Softwarové inženýrství ve vztahu k webovým aplikacím

Oblasti softwarového inženýrství a vývoje webových aplikací se vzájemně stále více propojují, což reflekтуje výrazný posun směrem k aplikacím založeným výhradně na webové platformě. Dříve byly aplikace typicky vytvářeny pro specifické operační systémy nebo hardwarové architektury, dnes se však často upřednostňuje vývoj webových aplikací přístupných prostřednictvím běžného internetového prohlížeče z libovolného zařízení či místa na světě. Přestože web nebyl původně zamýšlen jako univerzální softwarová platforma, postupně se stává preferovaným řešením pro vývoj většiny uživatelských aplikací (Mikkonen & Taivalsaari, 2010).

Hlavním cílem softwarového inženýrství je zajistit, aby procesy návrhu, vývoje, testování a údržby softwaru byly efektivní, systematické a dlouhodobě udržitelné. Je ale důležité rozlišovat rozdíl mezi softwarovým inženýrstvím a samotným programováním. Psaní kódu může představovat 10 % nebo méně času softwarového inženýra. Někdo, kdo tráví většinu času generováním kódu, je vhodněji nazýván programátorem. Softwarové inženýrství je širší a systematický přístup, který zahrnuje procesy, metodologie a řízení celého životního cyklu softwaru. Pouhé psaní kódu podle návrhu softwarového inženýra není inženýrskou činností (Laplante & Kassab, 2022).

#### Přístupy k psaní kódu

##### Tradiční přístup

Klasický způsob vývoje webových aplikací spočívá v přímém psaní zdrojového kódu v programovacích jazycích, jako jsou JavaScript (dále jako JS), Python, Java nebo Hypertext Preprocessor (dále jako PHP<sup>1</sup>). Tento přístup poskytuje vývojářům vysoký stupeň autonomie při definování architektury a logiky aplikace, což je zásadní pro projekty kladoucí důraz na zabezpečení, optimalizaci výkonu a přizpůsobivost. Avšak nároky na znalosti a odborné zkušenosti jsou v tomto případě značně vysoké. Nevhodou je proto zejména vyšší časová náročnost a s tím související vyšší finanční investice do celého vývojového procesu (Ritesh et al., 2023).

---

<sup>1</sup> PHP znamenalo původně Personal Home Page, ale momentálně se využívá spíše jako zkratka ve výše uvedeném smyslu.

## **Low code/no code přístup**

Moderní alternativou klasických vývojových metod jsou low-code a no-code platformy. Tyto nástroje nabízejí vizuální rozhraní a předem definované komponenty, čímž snižují potřebu manuálního programování. Low-code platformy jsou zaměřeny především na profesionální vývojáře, kteří chtějí urychlit vývoj bez zásadního omezení flexibility. Naopak no-code nástroje, jako Webflow či Bubble, jsou orientovány na uživatele bez technického vzdělání a umožňují rychlou tvorbu aplikací téměř bez psaní kódu (Rokis & Kirikova, 2023).

Webové systémy pro správu obsahu neboli web Content Management Systems (dále jen CMS), jako WordPress, patří do této kategorie právě díky schopnosti vytvářet digitální produkty bez nutnosti hlubších programátorských znalostí. Ačkoli CMS platformy nejsou vždy explicitně označovány jako low-code nástroje, sdílejí jejich základní principy a cíle jako například minimalizaci technické bariéry a usnadnění tvorby obsahu i složitějších digitálních produktů pro netechnické uživatele (Yermolenko & Golchevskiy, 2021).

## **Přístup s použitím umělé inteligence**

Nejnovější vývoj v oblasti webových aplikací charakterizuje intenzivní zapojení Artificial Intelligence (dále jen AI), a to buď jako podpůrného nástroje, nebo přímo aktivního účastníka vývojového procesu. AI nachází uplatnění zejména v automatizovaném generování zdrojového kódu, návrhu uživatelských rozhraní či automatizaci testovacích úloh. Díky dostupnosti nástrojů určených ke generování kódu, jako jsou GitHub Copilot nebo ChatGPT, mohou vývojáři snadněji a rychleji řešit technické problémy a vytvářet aplikace pouze na základě stručných instrukcí formulovaných v přirozeném jazyce (Upadhyaya, 2024).

Role AI zde ovšem nespočívá v prosté substituci lidských vývojářů, ale zejména v rozšíření jejich dovedností a kapacit. Tímto způsobem vzniká nový hybridní přístup, ve kterém se lidská kreativita synergicky spojuje s výpočetními možnostmi a predikčními schopnostmi moderních metod strojového učení, což umožňuje vývoj kvalitnějších a inovativnějších softwarových produktů (Jadhav & Gholve, 2024).

### **1.1.2 Historie vývoje webových aplikací**

Na počátku této kapitoly je nezbytné zasadit vývoj webových aplikací do širšího historického rámce a poskytnout ucelený přehled hlavních vývojových milníků. Ačkoli se dnes webové aplikace staly nedílnou součástí digitální infrastruktury a každodenního života, jejich vývojové kořeny sahají jen několik desetiletí zpět. Významné technologické i společenské proměny nastaly zejména po roce 2000, kdy došlo k rychlému rozvoji internetu a souvisejících nástrojů. V rámci této kapitoly bude vývoj webových aplikací strukturován do tří klíčových etap, které reflektují zásadní změny v architektuře, funkcionalitě i míře uživatelské interakce. Každá z těchto fází je spojena s konkrétním technologickým paradigmatem, jež formovalo podobu webových řešení dané éry.

Tyto etapy zahrnují:

- Web 1.0 (Read-only web).

- Web 2.0 (Read-write web).
- Web 3.0 (Portable and Personal web).

## **Web 1.0**

Vznik Webu 1.0 se datuje zhruba do období konce 90. let 20. století a počátku 21. století. Tato éra představuje první fázi internetu, která byla primárně charakterizována statickými webovými stránkami označovanými také jako read-only web (Ritesh et al., 2023).

Webové stránky typické pro období Webu 1.0 byly v zásadě statickými dokumenty, které obsahovaly převážně textový obsah a statické obrázky. Chyběla jim jakákoli forma animací nebo interaktivnosti; jejich hlavním účelem bylo jednoduché sdílení informací a distribuce dokumentů bez možnosti zpětné vazby nebo uživatelského zásahu do jejich podoby. Navigace mezi těmito stránkami byla řešena pomocí jednoduchých hypertextových odkazů. Každé kliknutí znamenalo načtení nové stránky ze serveru, přičemž veškerá komunikace probíhala synchronně. V této fázi vývoje internetu neexistovala potřeba asynchronního přenosu dat mezi prohlížečem a serverem (Taivalsaari & Mikkonen, 2011).

Web 1.0 fungoval převážně jako jednosměrné médium, které bylo vlastněno a řízeno vydavateli obsahu. Uživatelé se omezovali na roli pasivních konzumentů a interakce byla takřka nulová. Web v této podobě připomínal rozsáhlou digitální knihovnu, zaměřenou na přístup k informacím, nikoliv na jejich sdílení či úpravu. Počátky samotného webu sahají na začátek 90. let, kdy byl v prostředí CERNu vyvinut jako prostředek pro snadné a sjednocené sdílení informací. Hypertext jako klíčový nástroj pro propojování dokumentů umožnil vědcům jednoduchý přístup k datům z různých zdrojů (Anttonen et al., 2011).

Z technologického hlediska Web 1.0 stál na třech hlavních pilířích. Prvním byl Uniform Resource locator (dále jen URL), který umožňoval jednoznačnou identifikaci a odkazování na dokumenty v rámci sítě. Druhým byl HyperText Markup Language (dále jen HTML), tedy jazyk pro strukturování obsahu webových stránek, jenž definoval, jak má být obsah zobrazen. A třetím Hypertext Transfer Protocol (dále jen HTTP), protokol zajišťující komunikaci mezi klientem a serverem na základě jednoduchého modelu požadavku a odpovědi (Jazayeri, 2007).

V raném stádiu vývoje byla interaktivita webu značně omezená. Přestože se na některých stránkách začaly objevovat jednoduché formuláře, většina funkcionality byla statická a neumožňovala složitější operace. Na toto omezení mělo vliv i tehdejší pomalé internetové připojení, které nebylo schopno efektivně přenášet rozsáhlejší nebo dynamičtější obsah (Ritesh et al., 2023).

Období Webu 1.0 položilo základy pro další vývoj a transformaci internetu v interaktivnější a aplikativně bohatší prostředí, které známe dnes. Postupné uvědomění si potenciálu serverové strany pro dynamické generování stránek a následný vývoj technologií jako JavaScript a AJAX vedly k přechodu k Web 2.0.

## **Web 2.0**

Web 2.0 představuje vyspělejší generaci internetu navazující na éru Webu 1.0 a zhruba pokrývá období po přelomu tisíciletí. Hlavním mezníkem tohoto vývojového stádia bylo zavedení personalizovaného obsahu, uživatelských profilů a systémů, které umožnily vytvářet individuálně přizpůsobené online prostředí. V této souvislosti lze Web 2.0 označit jako read-write web (Ritesh et al., 2023).

Na rozdíl od statických stránek typických pro Web 1.0 přinesl Web 2.0 důraz na interaktivitu a dynamický obsah, změnil se způsob navrhování webových stránek a s rostoucím významem mobilních zařízení se prosadil responzivní design. Webové aplikace se staly citlivější na uživatelské vstupy a bylo možné aktualizovat jednotlivé prvky rozhraní bez nutnosti znovunačtení celé stránky (Anttonen et al., 2011).

S nástupem Webu 2.0 se zásadně proměnil způsob zapojení uživatelů, kteří se z pasivních příjemců informací stali jejich aktivními tvůrci. Objevily se nové formy využití internetu, jako jsou sociální sítě, blogy či platformy umožňující simultánní sdílení a spolupráci na obsahu. Uživatelé začali interagovat s obsahem prostřednictvím komentářů, hodnocení nebo sdílení médií, čímž se výrazně proměnila role publika v digitálním prostoru (Jazayeri, 2007).

Použití technologie Asynchronous JavaScript and XML (dále jen AJAX) představovalo významný mezník ve zvyšování kvality uživatelského zážitku. Díky této technologii bylo možné provádět aktualizace obsahu v reálném čase, přičemž došlo k oddělení přenosu dat od vizuálního rozhraní aplikace. Tato asynchronní architektura vytvořila základ pro interaktivnější a plynulejší webové aplikace. Zásadní roli zde sehrál i rostoucí význam client side JS<sup>2</sup> jehož širší využití umožnilo vznik Rich Internet Applications neboli komplexnějších webových řešení využívajících technologie jako Flash, Silverlight či Java applety. Vybrané aplikace, byť často závislé na externích plug-inech, významně posílily možnosti webového rozhraní a přiblížily jeho funkcionality tradičnímu desktopovým prostředím (Ritesh et al., 2023).

S rostoucím významem modularity a znova použitelnosti se Web 2.0 vyznačoval nástupem standardizovaných součástí. Vývojáři začali běžně nasazovat opakovaně použitelné prvky, jako jsou vyhledávací moduly, nástroje pro syndikaci<sup>3</sup> obsahu (například Really Simple Syndication) a funkce pro tagování obsahu. Zejména tagování hrálo klíčovou roli při zvyšování úrovně personalizace a efektivity navigace v rámci digitálního prostoru. Typickými aplikacemi Webu 2.0 se staly: Wikipedie – encyklopédie složená z příspěvků uživatelů, blogy, Flickr – web pro sdílení fotografií, MySpace – jedna z prvních sociálních sítí či Google Docs – document manager. Tyto příklady mají všechny něco společného: interaktivitu, sdílení a aktivní účast uživatelů (Jazayeri, 2007).

---

<sup>2</sup> Java script na straně klienta.

<sup>3</sup> Zkrácení obsahu do novinek.

## **Web 3.0**

Hlavní vizí Webu 3.0 je decentralizovaný internet, ve kterém se klade důraz na uživatele jako plnohodnotné vlastníky svých dat. V kontextu webových aplikací to znamená přechod k decentralizovaným aplikacím, které nejsou závislé na centrálních serverech a využívají technologie jako blockchain nebo peer-to-peer sítě. Tento směr lze historicky vidět jako pokus o návrat k původní myšlence otevřeného a distribuovaného webu, kde je posílena autonomie jednotlivých uživatelů. Termín se začal objevovat v roce 2013 a web 3.0, jako takový, trvá dodnes. Lze ho označit za portable and personal web (Goel et al., 2022).

Dopady Webu 3.0 na webové aplikace mohou být významně transformační. Architektura se decentralizuje: aplikace fungují na distribuovaných sítích, což omezuje rizika cenzury a zajišťuje větší odolnost vůči výpadkům. Zároveň uživatelé získávají nad svými daty větší kontrolu a mohou rozhodovat o jejich zpřístupnění či komerčním využití. Vznikají tak nové ekonomické modely, v nichž je pomocí nově vzniklých technologií umožněna přímá monetizace obsahu a zavádění konceptů digitálního vlastnictví. Důležitými atributy Webu 3.0 je také transparentnost a bezpečnost, které jsou posilovány využitím blockchainu pro záznam transakcí a operací s daty. Paralelně se rozvíjí sémantický web, jehož ambicí je zvýšit strojovou srozumitelnost dat a položit základ inteligentnějším a kontextově citlivějším aplikacím (Ritesh et al., 2023).

Z dlouhodobého pohledu lze evoluci od statického Webu 1.0, přes interaktivní Web 2.0 až po decentralizovaný Web 3.0 vnímat jako kontinuální proces technologické a společenské inovace. Tento proces směřuje k otevřenějšímu, bezpečnějšímu a uživatelsky kontrolovanému digitálnímu prostoru. I když je Web 3.0 stále ve svých raných fázích a potýká se s překážkami v oblasti škálovatelnosti a masové adopce, jeho potenciál pro změnu standartního smýšlení v oblasti webových aplikací je značný. Již dnes existují dApps<sup>4</sup>, které tuto změnu demonstруjí v oblasti financí, sociálních sítí či správy digitální identity. Uvedené příklady naznačují posun směrem k webu, kde jsou decentralizace, suverenita uživatelů a nové formy interakce klíčovými prvky jeho architektury i filozofie.

S rostoucí složitostí webových aplikací se začaly objevovat první webové vývojové frameworky na straně klienta, které usnadňovaly organizaci kódu a vývoj interaktivních uživatelských rozhraní. Je důležité poznamenat, že fáze Webu 1.0, Webu 2.0 a Webu 3.0 se neprekryvají ostře a prvky všech těchto iterací se používají na webu dodnes. Nicméně trend směrem k bohatším a interaktivnějším webovým aplikacím se stal zřejmým s nástupem Webu 2.0.

Web se v průběhu několika posledních dekád postupně zapsal do systému jako univerzální aplikační platforma, která dnes slouží jako primární prostředí pro provoz celé řady softwarových řešení. Z prostředí určeného primárně pro prohlížení dokumentů se stal nástroj schopný zastávat roli tradičního operačního systému. Tento přerod byl umožněn jak

---

<sup>4</sup> Decentralizované aplikace.

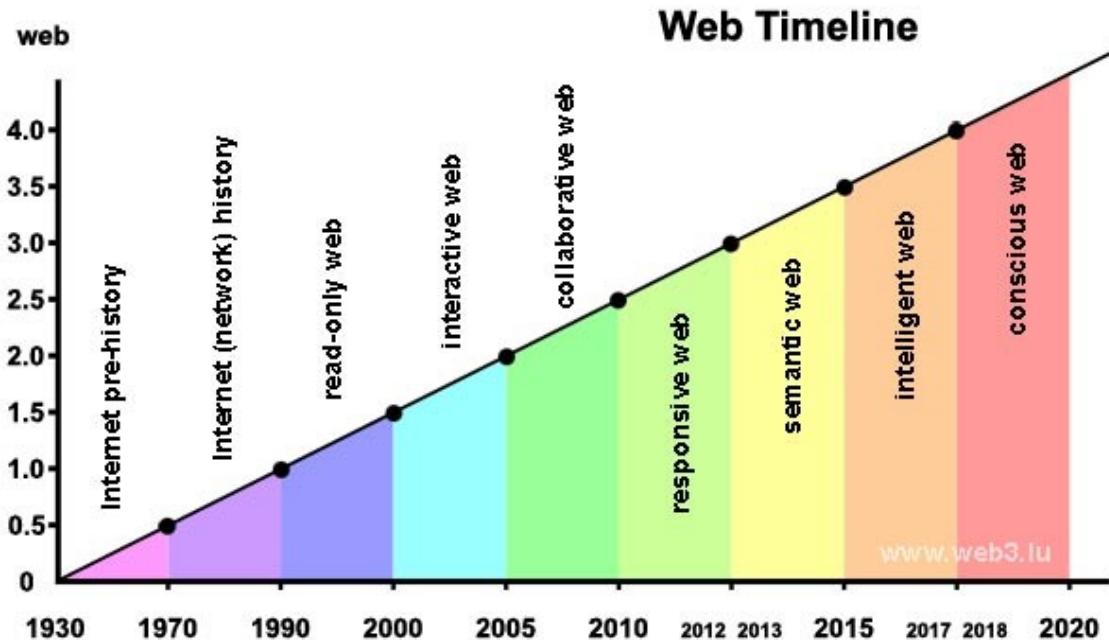
vývojem hardwaru a síťové infrastruktury, tak především díky zralosti webových standardů a nových technologií.

Transformaci webu v robustní platformu umožnil nástup nových standardů a technologií. HTML5 přinesl řadu funkcí zaměřených na podporu webových aplikací pro desktopové prostředí, včetně nativní podpory audia a videa, funkcí drag-and-drop a lokálního úložiště. Významnou roli sehrálo také rozšíření JavaScriptu a jeho ekosystému – včetně vývoje nových API, zlepšení syntaxe (ECMAScript) a zavedení nástrojů jako WebGL, které umožnily pokročilou grafiku přímo v prohlížeči. Díky těmto inovacím přestal být web omezován původními limity svého návrhu a začal se profilovat jako prostředí vhodné pro komplexní aplikace (Anttonen et al., 2011).

S rostoucí složitostí webových aplikací se začaly prosazovat specializované vývojové frameworky. V oblasti tvorby uživatelského rozhraní se rozšířily nástroje jako React, Angular a Vue, které usnadňují tvorbu dynamických rozhraní a Single-Page Aplikací (dále jen SPA). Na straně zpracování aplikační logiky a práce s databázemi se rozšířily frameworky jako Ruby on Rails, Django, SpringBoot či Laravel, které zjednodušily práci s databázemi i serverovou logikou. Tyto frameworky přinesly standardizaci komponent a nástrojů, čímž přispěly k vyšší efektivitě a konzistenci vývoje (Ritesh et al., 2023).

Jednou z klíčových výhod webových aplikací je jejich okamžité nasazení v globálním měřítku bez nutnosti instalace nebo ručních aktualizací. Oproti tradičnímu softwaru distribuovanému v binární podobě tak představují zásadní posun v přístupu k šíření a údržbě aplikací. Budoucnost webu jako aplikační platformy je spojena s konceptem Webu 3.0, který umožní inteligentnější zpracování informací a společně s vývojem v oblasti cloud computingu, zlepšujícími se schopnostmi webových prohlížečů a výkonnější síťovou infrastrukturou dále posiluje pozici webu jako dominantního prostředí pro vývoj a distribuci aplikací (Taivalsaari & Mikkonen, 2011).

Web se tak z prostředí určeného pro pasivní konzumaci obsahu vyvinul v komplexní platformu, která v mnoha případech nahrazuje tradiční desktopové aplikace. Tento vývoj byl umožněn kombinací technologických inovací, vznikem standardů a nástrojů podporujících vývoj a zrání samotného webového ekosystému.



Obrázek 1 - Časová osa historie webu (Barning, 2013)

### 1.1.3 Rozdělení vývoje webové aplikace

Moderní vývoj webových aplikací zahrnuje široké spektrum činností, které lze rámcově rozdělit na dvě hlavní oblasti: frontend a backend. Každá z těchto oblastí představuje samostatnou vrstvu systému s vlastními technologiemi, metodikami a odpovědnostmi. Zatímco frontend se zaměřuje na to, co uživatel přímo vidí a ovládá, backend zajišťuje funkčnost na pozadí – od zpracování dat až po logiku aplikace. Jejich vzájemná provázanost je klíčová pro tvorbu stabilních, bezpečných a uživatelsky přívětivých řešení. Následující část se podrobně venuje jejich vymezení, specifikům a nárokům na vývojáře.

#### Frontend

Frontendový vývoj označuje proces návrhu a implementace uživatelského rozhraní, tedy všech vizuálních a interaktivních prvků, se kterými uživatel přichází do styku prostřednictvím webového prohlížeče. Vývojáři na straně klienta pracují s technologiemi jako HTML pro strukturování obsahu, Cascading Style Sheet (dále jako CSS) pro jeho stylizaci a JavaScript pro implementaci dynamického chování. Důraz je kladen na použitelnost, vizuální konzistenci a responzivitu napříč různými zařízeními.

Frontendoví vývojáři se zabývají nejen estetickým návrhem aplikace, ale i implementací komponent, které umožňují interakci s uživatelem a přispívají k celkové uživatelské zkušenosti. V této oblasti se od vývojáře očekává schopnost efektivní práce s návrhovými vzory, přístupnost, optimalizace výkonu a zvládnutí rychlého tempa vývoje s ohledem na časté aktualizace knihoven, frameworků a nástrojů (Rahaman et al., 2022).

## **Backend**

Backendový vývoj se zaměřuje na logiku aplikace, její datovou vrstvu a komunikaci se serverem. V této části se zpracovávají uživatelské požadavky, uchovávají a načítají data z databáze a generují odpovědi pro klientskou část aplikace. Vývoj probíhá zpravidla v jazycích jako PHP, Python nebo Node.js a využívá relační či nerelační databázové systémy např. MySQL nebo MongoDB.

Úkolem backendového vývojáře je navrhovat stabilní aplikační architekturu, implementovat rozhraní pro výměnu dat a zajistit bezpečný, výkonný a škálovatelný provoz aplikace. Tato část vývoje rovněž zahrnuje správu uživatelských účtů, autentizaci, autorizaci a optimalizaci práce s databázemi. Backendová logika je často neviditelná, ale nezbytná pro správnou funkčnost celého systému (Rahaman et al., 2022).

Fullstackový vývoj představuje přístup, v němž jeden vývojář ovládá jak frontendové, tak backendové technologie a je schopen samostatně vyvíjet plně funkční aplikace od návrhu uživatelského rozhraní po serverovou logiku a práci s databázemi. Tato role vyžaduje široké technologické znalosti, schopnost rychle přepínat mezi kontexty a pochopení souvislostí napříč celým vývojovým cyklem (Rahaman et al., 2022).

Dobré porozumění rozdílům mezi frontendem a backendem je zásadní nejen pro samotné vývojáře, ale také pro efektivní spolupráci v rámci vývojových týmů. Harmonická integrace obou částí přispívá k lepší kvalitě výsledné aplikace, její udržovatelnosti i dlouhodobé škálovatelnosti.

### **1.1.4 Průběh vývoje webové aplikace**

Při popisu následujících kroků byl čerpán inspirační rámcem z publikace Learning Web Development od Semmyho Purewala. Purewal ve své práci představuje vývoj webových aplikací jako sled praktických fází, které mají vymezenou strukturu i posloupnost. Text se této koncepce drží a jednotlivé kroky zpracovává formou popisného přehledu s důrazem na srozumitelnost a návaznost jednotlivých činnost (Purewal, 2014).

#### **Určení pracovního postupu (Workflow)**

Tento krok je základem pro efektivní vývoj webové aplikace. Zahrnuje výběr a používání nástrojů, které vývojář potřebuje pro psaní, správu a testování kódu. Klíčovými komponentami tohoto kroku jsou textové editory, systémy pro správu verzí a vývojářské nástroje webového prohlížeče. Textový editor je primární nástroj pro interakci s kódem, proto je důležité, aby usnadňoval jeho vytváření a úpravy. Sammy Purewell uvádí příklady jako Sublime Text, Emacs a Vim. Systém pro správu verzí, jako je Git, umožňuje sledovat změny v kódu, vracet se k předchozím verzím a spolupracovat s ostatními vývojáři. Dále autor uvádí, že webové prohlížeče, jako je Chrome, slouží nejen k zobrazení aplikace, ale i jako vývojářský nástroj pro interakci s kódem a jeho ladění (Purewal, 2014).

## **Utvoření struktury (Structure)**

V tomto kroku se definuje základní kostra webové stránky pomocí Hyper Text Markup Language. HTML používá tagy k přidání struktury a významu obsahu. Tato struktura se reprezentuje jako Document Object Model (dále jako DOM), hierarchický strom prvků. Správná sémantická struktura je důležitá pro přístupnost a Search Engine Optimization (dále jako SEO). Za účelem ověření správnosti HTML kódu je jako další součást tohoto kroku doporučena HTML validace (Purewal, 2014).

## **Určení stylu (Style)**

Po vytvoření základní struktury aplikace následuje fáze definice vzhledu, který je realizován pomocí jazyka CSS neboli Cascading Style Sheets. Stylování probíhá prostřednictvím tzv. rulesetů – seskupených pravidel aplikovaných na vybrané HTML elementy. Každý ruleset obsahuje selektor a sadu deklarací, které určují vizuální vlastnosti jako barvu, písmo, rozvržení či odsazení. V současném vývoji je kláden důraz na responzivní design, který umožňuje flexibilní přizpůsobení rozvržení a velikostí prvků podle rozlišení zařízení. Běžně se využívají techniky jako media queries, procentuální jednotky nebo flexibilní mřížky: Flexbox a CSS Grid. Tato opatření zajišťují, že obsah zůstává čitelný a přístupný napříč platformami (Purewal, 2014).

## **Zaručení interaktivity (Interactivity)**

Po určení stylu aplikace přichází na řadu přidání interaktivity. Ta je realizována pomocí jazyka JavaScript, který běží přímo v prostředí webového prohlížeče. JavaScript umožňuje obohatit webovou aplikaci o dynamické prvky a reagovat na chování uživatele bez nutnosti znova načítat celou stránku. Díky tomu se výrazně zvyšuje plynulost a interaktivita prostředí. V rámci vývoje slouží JavaScript především k řízení chování rozhraní, zpracování vstupů a navazování komunikace se serverem, čímž překračuje hranice statického webu a přibližuje aplikaci zkušenosti typické pro desktopový software (Purewal, 2014).

## **Komunikace aplikace s klientem (The Bridge)**

Moderní webové aplikace vyžadují efektivní výměnu dat mezi klientem a serverem. Tento proces je zajištěn pomocí formátu JavaScript Object Notation (dále jako JSON), který slouží jako standard pro strukturovanou výměnu informací v podobě srozumitelné pro člověka i stroj. K samotnému přenosu se využívá technologie Asynchronous JavaScript And XML (dále jako AJAX), jež umožňuje asynchronní komunikaci bez nutnosti kompletního obnovení stránky. Díky tomu může aplikace pružně reagovat na uživatelské vstupy a aktualizovat pouze relevantní části obsahu. Tato forma komunikace významně přispívá k plynulosti, modularitě a celkové uživatelské přívětivosti webových řešení (Purewal, 2014).

## **Nastavení serverové logiky (The Server)**

Následuje vytvoření logiky na straně serveru. Serverová strana aplikace zajišťuje reakci na příchozí požadavky, zpracování dat a přípravu odpovědí. Tento segment vývoje klade důraz na efektivní zpracování síťové komunikace, práci s databázemi a správné směrování požadavků na jednotlivé části systému. Zároveň je zde klíčové zabezpečení aplikace, ochrana dat a optimalizace výkonu. Vývoj serveru vyžaduje schopnost pracovat s rozhraními,

definovat strukturu backendu a zajistit plynulou součinnost s klientskou částí aplikace. V této fázi se běžně využívají prostředí jako Node.js které nabízí rozsáhlou sadu nástrojů pro správu serverové logiky a komunikaci s databázemi (Purewal, 2014).

### **Zajištění datového uložiště (The Data Store)**

Trvalé uchovávání dat je zajišťováno prostřednictvím databázových systémů. Webové aplikace v současnosti často využívají nerelační databáze typu NoSQL, které poskytují vysokou flexibilitu a škálovatelnost. Například dle Semmyho Purewela MongoDB, jako dokumentově orientovaná databáze, uchovává data ve formátu podobném JSON, čímž usnadňuje jejich integraci do JavaScriptových aplikací. Pro zjednodušení práce s touto databází je nasazována knihovna Mongoose, která zajišťuje mapování dokumentů na objekty a poskytuje nástroje pro validaci a manipulaci s daty (Purewal, 2014).

### **Nasazení a provoz (The Platform)**

Aby byla webová aplikace přístupná uživatelům na internetu, je nutné ji nasadit na hostingovou platformu. Využívají se služby typu Platform-as-a-Service (dále jako PaaS), například Heroku nebo Cloud Foundry. Tyto platformy poskytují kompletní prostředí pro běh a správu aplikací v cloudu, včetně serverů, operačních systémů a potřebného softwaru. Nasazení obvykle zahrnuje konfiguraci aplikace pro danou platformu např. nastavení portů pomocí proměnných prostředí a správu systémových závislostí. Tyto platformy často umožňují propojení s databázemi a nabízí nástroje pro sledování výkonu a logování (Purewal, 2014).

### **Dokončení a údržba (The Application)**

Závěrečná fáze vývoje se zaměřuje na dlouhodobou udržitelnost a kvalitu aplikace. Důležitou součástí je refactoring, tedy úprava kódu za účelem zlepšení jeho struktury a čitelnosti bez změny funkcionality. Pro organizaci kódu se často uplatňuje určitý architektonický vzor. Jedním z takových vzorů je Model-View-Controller, který rozděluje aplikaci do tří základních komponent podle jejich odpovědnosti. Komunikace se světem je realizována pomocí rozhraní pro řízení serverových Application Programming Interface (dále jako API), které definuje pravidla pro přístup ke zdrojům skrze standardní HTTP metody. Důsledná obsluha chyb a logování událostí zajišťují robustnost a spolehlivost celého systému (Purewal, 2014).

Vývoj webové aplikace představuje komplexní proces, který propojuje řadu specializovaných oblastí – od návrhu struktury a stylování rozhraní přes programování interaktivnosti až po implementaci serverové logiky a databázové vrstvy. Úspěšné zvládnutí jednotlivých kroků vyžaduje nejen technické znalosti, ale i porozumění principům efektivní spolupráce mezi frontendovou a backendovou částí systému. Vzhledem k rychlému vývoji technologií a měnícím se požadavkům uživatelů je nezbytné klást důraz na udržitelnost, modularitu a škálovatelnost řešení. Správně navržený vývojový proces tak nejen usnadňuje údržbu aplikace, ale rovněž vytváří pevný základ pro její další rozšiřování a adaptaci na budoucí potřeby.

## 1.1.5 Časté problémy s vývojem webových aplikací

Vývoj fullstackových webových aplikací s sebou nese řadu komplikací. Ty se netýkají pouze samotné technologické náročnosti, ale také rychlého vývoje celého odvětví. Vysoké nároky na kvalitu, stabilitu a bezpečnost činí z této oblasti neustále se vyvíjející výzvu. Tato část práce vychází z poznatků uvedených ve studii publikované ve World Journal of Advanced Engineering Technology and Sciences, která mapuje nejčastější překážky spojené s vývojem fullstackových webových aplikací:

- **Chyby v kódu:** Pokud nejsou na začátku projektu přesně vymezeny požadavky, dochází k uspěchanému vývoji. Ten často končí detaily, které se zdají být drobné – například chybějící středník nebo nesprávně uzavřený blok – ale ve výsledku mohou způsobit zásadní nestabilitu aplikace.
- **Rychlá evoluce technologií:** Neustálé přizpůsobování novým frameworkům, knihovnám a jazykům zvyšuje tlak na průběžné vzdělávání. Výběr vhodného nástroje se v tomto kontextu stává obtížným, protože množství možností často vyvolává zmatek a váhavost.
- **Předčasná optimalizace:** Přestože snaha o efektivní provoz může působit jako rozumný krok, často probíhá předčasně a poté vede ke zbytečnému snížení kvality kódu.
- **Úroveň abstrakce:** Příliš komplikovaná struktura snižuje čitelnost a zpomaluje efektivitu práce s kódem. Naopak příliš nízká úroveň abstrakce vede ke zbytečnému opakování a obtížné údržbě. Rovnováha mezi těmito póly je proto klíčová.
- **Kompatibilita mezi technologiemi:** Různá syntaxe, API či verze využívaných technologických řešení mohou být zdrojem nesnází. Jejich vzájemná nekompatibilita často komplikuje integraci jednotlivých komponent.
- **Přenos dat:** Přenos dat mezi systémy je náročný zejména tehdy, pokud formáty či protokoly nejsou vzájemně kompatibilní.
- **Údržba a podpora:** Až v ostrém provozu se odhalují problémy, jako je nízká rychlosť načítání, bezpečnostní slabiny nebo nedostatečný výkon. Tyto problémy vyžadují konzistentní monitoring a pravidelnou aktualizaci.
- **Bezpečnost:** Moderní webové aplikace pracují s citlivými daty, a proto musí být chráněny před útoky, nedostatečnou autentizací či špatným nastavením oprávnění. Právě důraz na ochranu informací tvoří dnes základní kámen důvěryhodné aplikace (Attri & Pathania, 2025).

## 1.2 Úvod do umělé inteligence a rozsáhlých jazykových modelů

### 1.2.1 Umělá inteligence

Inteligence bývá obecně chápána jako schopnost učit se a aplikovat vhodné techniky pro řešení problémů a dosažení cílů s ohledem na aktuální kontext v proměnlivém a nejistém světě. Tato schopnost není vlastní všem automatizovaným systémům. Například plně naprogramovaný tovární robot může být přesný, flexibilní a konzistentní, avšak postrádá atribut inteligence – schopnost adaptace a učení.

Termín Artificial Intelligence byl poprvé použit v roce 1955 profesorem Stanfordské univerzity Johnem McCarthym, který AI definoval jako „vědu a inženýrství vytváření inteligentních strojů“. Zatímco dřívější výzkumy se zaměřovaly na programování strojů tak, aby se chovaly chytré – například hrály šachy – dnešní důraz je kladen především na schopnost strojů učit se. Tato změna paradigmat od předem daného chování ke schopnosti samostatného učení přiblížila stroje možnostem lidského myšlení (Manning, 2020).

### **Autonomie a adaptace**

Zásadní roli v rámci AI hrají autonomní systémy, které dokážou samostatně plánovat a rozhodovat o krocích vedoucích k dosažení specifikovaného cíle bez nutnosti detailního dohledu. Typickým příkladem může být nemocniční doručovací robot, jenž musí zvládnout samostatnou navigaci v rušných chodbách nemocnice. V kontextu AI pojmem autonomie neoznačuje politickou nebo biologickou samosprávu, ale schopnost řídit své vlastní chování na základě dat a vnitřních rozhodovacích mechanismů.

### **Strojové učení**

Machine Learning (dále jako ML), jako další důležitá složka umělé inteligence, zkoumá, jak mohou počítačoví agenti zlepšovat své schopnosti vnímání, rozhodování, uvažování nebo jednání na základě dat či zkušeností. Strojové učení čerpá z řady vědních oborů, včetně informatiky, statistiky, psychologie, neurověd, ekonomie a teorie řízení (Manning, 2020).

Přístupy ke strojovému učení:

- Učení s učitelem (supervised learning): model se učí předpovídat výstupy na základě předem označených dat, například rozpoznávání plemen psů.
- Učení bez učitele (unsupervised learning): model vytváří predikční úkoly bez nutnosti anotovaných dat, například předpovídání slov ve větě.
- Učení posilováním (reinforcement learning): model získává odměny za akce vedoucí k cíli, bez předem definovaných příkladů správného chování.

### **Hluboké učení**

Zvláštní zmínu si zaslouží Deep Learning, které využívá rozsáhlé vícevrstvé umělé neuronové sítě pracující s reálnými číselnými reprezentacemi. Tyto sítě svou strukturou částečně připomínají hierarchii neuronů v lidském mozku. Hluboké učení je v současnosti nejúspěšnější metodou strojového učení, protože umožňuje lepší generalizaci i při práci s malými množstvími dat a efektivněji škáluje na rozsáhlé datové soubory a výpočetní kapacity (Liang et al., 2023).

### **Algoritmy a Software 2.0**

AI systémy využívají algoritmy – přesně definované kroky, které specifikují, jak dospět k určitému výsledku. Nicméně ve většině pokročilých systémů jsou algoritmy používány jen pro dílčí části (např. výpočet odměn či metoda učení), zatímco podstatná část chování vzniká učením z dat. Tento posun od klasického programování k adaptivním systémům nazval Stanfordský absolvent Andrej Karpathy výrazem Software 2.0.

## **Formy a směry AI:**

- Narrow AI: systémy zaměřené na konkrétní úkoly, jako je rozpoznávání řeči nebo obličeje.
- Obecná umělá inteligence: Artificial General Intelligence má za cíl vytvořit univerzální inteligenční stroje schopné chápout kontext a jednat ve složitých situacích, např. pro účely přirozené interakce s lidmi.
- Na člověka zaměřená AI: Human-Centered AI je umělá inteligence navrhována tak, aby rozšiřovala lidské schopnosti, reagovala na společenské potřeby a inspirovala se lidským chováním. Příkladem může být vývoj robotických asistentů pro seniory (Jurafsky & Martin, 2025).

Závěrem lze konstatovat, že uvedený přehled klíčových pojmu, principů a vývojových trendů v oblasti umělé inteligence poskytuje pevný teoretický základ pro porozumění jejímu využití při vývoji webových aplikací. Následující kapitola se proto zaměří na konkrétní podoblast AI, velké jazykové modely, a jejich roli ve zjednodušení a automatizaci vývojového procesu aplikací.

### **1.2.2 Velké jazykové modely**

Momentálním technologickým standardem a základem Large Language Modelu (dále jako LLM) je architektura zvaná Transformer. Místo aby zpracovávaly text slovo po slově, jako to dělaly starší programy, Transformery se zaměřují na vztahy mezi různými slovy v celém textu najednou. Funguje to obdobně jako skládání puzzlí. Transformer se snaží pochopit, jak spolu jednotlivá slova souvisejí, aby vytvořily smysluplný text – obraz (Xiao & Zhu, 2025).

První popis architektury transformer byl proveden týmem společnosti Google v roce 2017 v odborném článku s názvem „Attention is all you need“, kde autoři představili zásadní průlom ve zpracování sekvenčních dat představením právě technologie transformeru. Klíčovou součástí Transformerů je mechanismus pozornosti (attention). Ten umožnuje modelu věnovat různou míru důležitosti různým slovům v závislosti na kontextu. Když například generuje větu, model s pozorností ví, která slova z předchozí části věty jsou nejdůležitější pro výběr dalšího slova (Vaswani et al., 2017).

### **Architektury**

Existují různé typy architektur LLM, které se liší způsobem, jakým používají pozornost a jak jsou propojeny jejich základní stavební bloky:

- Causální dekodér (Causal Decoder): Tento typ architektury generuje text postupně, slovo po slově, a každé další slovo závisí pouze na slovech, která mu předcházela. Je to jako když píšete dopis – každé další slovo navazuje na to, co jste napsali předtím. Většina známých LLM, jako je GPT-3, používá tuto architekturu.
- Enkodér-dekodér (Encoder-Decoder): Tato architektura nejprve přečte celý vstupní text – enkodér a vytvoří jeho vnitřní reprezentaci. Poté tato reprezentace slouží

dekodéru k vygenerování výstupního textu. Tento typ se často používá pro úkoly jako překlad jazyků.

- Prefixový dekodér (Prefix Decoder): Tento typ je také známý jako ne-kauzální dekodér. Na rozdíl od kauzálního dekodéru nezávisí při výpočtu pozornosti striktně na minulých informacích a může brát v úvahu informace z obou stran.
- Směs expertů (Mixture-of-Experts): Toto je varianta, která obsahuje několik nezávislých expertů a router, který směruje vstupní slova k těmto expertům. Při každém vstupu se aktivuje pouze několik expertů najednou, což zvyšuje efektivitu a umožňuje vytvářet velmi velké modely (Xiao & Zhu, 2025).

## Proces učení

LLM se učí ve dvou hlavních fázích: pre-training a fine-tuning. V pre-tréninkové fázi je model vystaven obrovskému množství textových dat a učí se předvídat další slovo v sekvenci. Tento proces se nazývá self-supervised learning, model se učí sám z dat, aniž by potřeboval explicitní značky nebo návody od lidí. Cílem je maximalizovat pravděpodobnost výskytu trénovacích dat. Představte si, že dáváte papouškovi číst miliony knih a on se postupně učí, jak slova obvykle následují za sebou (Jurafsky & Martin, 2025).

Během pre-tréninku se model naučí obecné zákonitosti jazyka, jako je gramatika, slovní zásoba a základní fakta o světě. Kromě předpovídání dalšího slova existují i další pre-tréninkové úkoly, které pomáhají modelu lépe porozumět jazyku:

- Maskování jazyka (Masked Language Modeling): Některá slova ve vstupním textu jsou nahrazena speciálním symbolem a model se učí původní slova obnovit na základě kontextu. Tento přístup využívá například model BERT – Bidirectional Encoder Representations from Transformers.
- Permutované jazykové modelování (Permuted Language Modeling): Pořadí slov ve vstupní sekvenci je náhodně zamícháno a model se učí předvídat původní pořadí (Xiao & Zhu, 2025).

Po pre-tréninku je model doladěn na menších, specifických datasetech pro konkrétní úkoly. Tento proces se nazývá fine-tuning. Například, pokud chceme LLM použít pro odpovídání na otázky, doladíme ho na datech obsahujících otázky a správné odpovědi. Během doladění jsou parametry modelu jemně upravovány dohlížející osobou, aby model dosahoval lepších výsledků na daném úkolu.

Důležitým aspektem doladění je alignment, jelikož je cílem, aby model byl nejen přesný, ale také užitečný, neškodný a odpovídal lidským preferencím. To se často provádí pomocí technik jako učení s lidskou zpětnou vazbou – Reinforcement Learning from Human Feedback (dále jako RLHF). Při aplikování této techniky hodnotí lidé různé odpovědi modelu na stejnou otázku a tyto preference se používají k dalšímu trénování modelu (Naveed et al., 2024).

## Tokenizace

Než model začne zpracovávat text, musí být text rozdělen na menší jednotky zvané tokeny. Tokenem může být slovo, část slova nebo dokonce jeden znaky, z tohoto důvodu je proces

nazýván tokenizace. Efektivní zpracování a rozdelení textu je důležité pro zvýšení výpočetního výkonu modelu. Některé modely používají subword tokenizaci, která rozděluje slova na menší části, aby lépe zvládala složitá slova a různé formy slov. Například slovo "nejdůležitější" by mohlo být rozděleno na "nej", "důležitěj", "ší" (Jurafsky & Martin, 2025).

## Generování textu

Když LLM dostane vstupní text, například otázku nebo začátek věty, použije naučené znalosti a vzory k vygenerování pokračování. Tento proces se obvykle děje autoregresivně, což znamená, že model generuje text slovo po slově, a každé nově vygenerované slovo se stává součástí kontextu pro generování dalšího slova. Při výběru dalšího slova model vypočítá pravděpodobnost všech slov v jeho slovníku a vybere to s nejvyšší pravděpodobností. Existují ale i strategie vzorkování, které zavádějí určitou míru náhody, aby generovaný text nebyl příliš předvídatelný.

Kromě textu mohou LLM pracovat i s dalšími typy dat, modalitami, jako jsou obrázky, videa a zvuk. Tyto modely se nazývají multimodal large language models. MLLM integrují informace z různých modalit, což jim umožňuje hlubší porozumění kontextu a generování inteligentnějších a rozmanitějších odpovědí. Existují různé způsoby, jak tyto modely konstruovat, ale ve většině případů se stále drží formule pre-trénink, doladění a prompting s multimodálními daty (Xiao & Zhu, 2025).

## Promptování

Promptování textu pro LLM funguje jako metoda, jak těmto modelům zadat specifický vstup nebo podnět tak, aby modely generovaly požadovaný výstup nebo provedly určitý úkol. Jinými slovy, prompt je celý textový vstup, který je poskytnut LLM. Prompt slouží jako kontext nebo podmínka, na jejímž základě model provádí predikci a generuje tak požadovaný text. Model maximalizuje pravděpodobnost vygenerování textu vzhledem k zadanému promptu (Jurafsky & Martin, 2025).

Prompty mohou sloužit buď jako přímé instrukce, nebo jako šablony pro specifické úkoly. Instruktivní prompt zadává modelu jasný pokyn, např. „Přeložte následující text z angličtiny do čínštiny: ‘The early bird catches the worm.’“ Naopak šablonovitý prompt využívá zástupné proměnné, které se následně doplňují konkrétními údaji – například: „Pokud {x\*}, tak jaké jsou tvé návrhy na zábavný víkend?“ Takové prompty umožňují variabilní a opakování použití při různých vstupních datech (Xiao & Zhu, 2025).

Další důležitou součástí techniky promptování je In-context learning. Učení se v kontextu představuje důležitý princip při práci s jazykovými modely. Umožňuje systému učit se přímo z textu zadaného v promptu, bez nutnosti dalšího trénování. Tento způsob využívá kontextové informace, například ukázky vstupů a výstupů, k tomu, aby model pochopil, jaký druh odpovědi se od něj očekává. Podle rozsahu těchto ukázků rozlišujeme několik přístupů:

- Zero-shot learning označuje situaci, kdy model plní úkol pouze na základě slovní instrukce bez jakýchkoli příkladů.
- One-shot learning zahrnuje jeden příklad vstupu a odpovídajícího výstupu.

- Few-shot learning využívá několik příkladů, čímž se zvyšuje šance na přesné splnění úkolu.

Při návrhu promptů existuje několik osvědčených strategií, které mohou výrazně zlepšit výkon vybraného LLM. Základním předpokladem je přesné a jednoznačné zadání úkolu, tudíž čím konkrétněji je problém popsán, tím pravděpodobněji model porozumí očekávání a zvolí odpovídající postup. U složitějších úloh, jež vyžadují hlubší úroveň uvažování, lze efektivity dosáhnout naváděním modelu k myšlení krok za krokem. Tato technika, známá jako Chain of Thought, se často realizuje pomocí formulací typu „Pojďme přemýšlet krok za krokem“, které modelu signalizují, že má nejprve strukturovaně analyzovat zadání a teprve poté formulovat výstup.

Dalším užitečným přístupem je doplnění promptu o referenční informace. V praxi to znamená, že do kontextu jsou vkládány relevantní texty, ukázky nebo dokumenty, které model využívá jako podpůrný zdroj. Dále hraje roli i samotné formátování promptu. Přehledná struktura s jasně oddělenými sekciemi nebo využití kódového stylu zápisu může napomoci srozumitelnosti zadání a přispět ke konzistentnějším výsledkům. Správně zvolený formát totiž modelu usnadňuje orientaci ve vstupu a tím i přesnější generování výstupu (Xiao & Zhu, 2025).

### **1.2.3 Technologie umělé inteligence využitelné při vývoji webových aplikací**

V této podkapitole se bude autor věnovat příkladům technologií založených na umělé inteligenci využitelných při vývoji webových aplikací.

#### **Strojové učení**

Algoritmy strojového učení jsou klíčové pro automatizaci analýzy dat a rozhodovacích procesů při vývoji webových stránek. Tyto algoritmy dokážou předvídat chování uživatelů, přizpůsobovat doporučení a automatizovat reakce zákaznických služeb, čímž snižují potřebu lidských zásahů a zajišťují hladší uživatelskou zkušenosť na webové stránce (Rustambek, 2023).

#### **Natural Language Processing**

Technologie přirozeného zpracovávání jazyka zásadním způsobem ovlivňuje kvalitu interakce mezi uživateli a webovými aplikacemi. Díky pokročilé schopnosti interpretovat a reagovat na přirozené uživatelské dotazy virtuální asistenti zajišťují efektivnější a přirozenější komunikaci, což významně zlepšuje uživatelskou zkušenosť.

#### **Analytika řízená umělou inteligencí**

Analytika řízená umělou inteligencí neboli AI-driven analytics může být součástí procesu získávání poznatků ze složitých souborů dat. Tato technologie umožňuje identifikovat vzorce, trendy a korelace v chování uživatelů a dává tímto způsobem vhled vývojářům do chování uživatelů, výkonnosti obsahu a návštěvnosti webových stránek z analytického

pohledu. Na základě těchto poznatků lze následně optimalizovat webové stránky a přizpůsobit marketingové strategie tak, aby se zvýšila angažovanost a spokojenost uživatelů.

### **Detekce hrozeb v reálném čase**

Další věc, kterou je umělá inteligence schopna zvyšovat zabezpečení webu, je automatická detekce a boj proti potenciální hrozbám v reálném čase. Tento proaktivní přístup pomáhá udržet důvěru uživatelů a chrání citlivé informace před kybernetickými hrozbami (Rustambek, 2023).

### **Computer vision**

Computer vision se stále častěji používá při vývoji webových stránek pro úlohy, jako je automatizované testování a optimalizace návrhu grafického uživatelského rozhraní. Stocco uvádí, že techniky počítačového vidění se používají k analýze a porozumění obrázkům a k automatizaci vytváření struktur HTML z ručně kreslených obrázků, což výrazně urychluje proces návrhu (Stocco, 2019).

### **Deep learning**

Deep learning, je obzvláště efektivní při řešení složitých úloh, jako je rozpoznávání obrazu a řeči. Tyto schopnosti jsou použitelné v moderních webových aplikacích, kde hluboké učení umožňuje automaticky extrahat data. Díky tomu lze například rozvíjet webová rozhraní s hlasovou asistencí, která jsou schopná porozumět požadavkům uživatelů na hlubší úrovni.

### **Automatizované testování**

Role umělé inteligence ve vývoji webových aplikací se rozšiřuje i do specializovanějších oblastí, jako je automatizované testování. AI může například zlepšit efektivitu a účinnost testování webových aplikací tím, že automatizuje generování testovacích případů z prototypů grafického uživatelského rozhraní. To nejen urychluje proces testování, ale také pomáhá rychleji identifikovat potenciální problémy (Stocco, 2019).

Ve výsledku lze říci, že v oblasti vývoje webových aplikací má umělá inteligence mnoho možných využití. Na některé z těchto využití se zaměří autor v následující kapitole práce.

# **2 Praktické zhodnocení přínosů vývoje webových aplikací pomocí umělé inteligence**

## **2.1 Použité nástroje pro praktickou část**

V této části práce jsou popsány veškeré klíčové technologie a nástroje, využité během praktických experimentů. Výběr těchto nástrojů byl veden především snahou o zajištění komplexnosti a pokrytí všech aspektů webového vývoje, od ručního programování až po automatizaci prostřednictvím umělé inteligence a použití standardizovaných CMS platform. Zvolená řešení umožňují důkladné porovnání efektivity, flexibility a kvality výsledných webových aplikací v různých přístupech.

### **Visual Studio Code**

Současný trh nabízí široké spektrum nástrojů určených k tvorbě webových stránek, včetně no-code a low-code řešení, jako jsou například Bubble, Lovable nebo Webflow. Tyto platformy kladou důraz na rychlosť, automatizaci a vizuální přístup, což je vhodné především pro méně technicky orientované uživatele. V kontextu této práce byl pro ruční i AI generované webové stránky nakonec zvolen editor Visual Studio Code (dále jako VSC), protože cílem práce bylo porovnat metody vývoje založené na přímé práci s kódem.

VSC představuje pokročilé vývojové prostředí, které umožňuje detailní kontrolu nad strukturou, syntaxí a kvalitou zdrojového kódu. Díky své otevřenosti a nástrojové podpoře lépe odpovídá požadavkům na metodologicky konzistentní vývojové zázemí. V porovnání s jinými plnohodnotnými editory, jako například Sublime text, jsem VSC zvolil pouze na základě mých předešlých zkušeností s prací v tomto prostředí. Momentálně nejnovější stabilní verzí VSC v době psaní této práce je verze 1.89.0.

### **ChatGPT**

Pro tuto práci jsem využil jazykový model ChatGPT ve verzi GPT-4.5 od společnosti OpenAI. Tento nástroj je schopen na základě textového zadání generovat části kódu v HTML, CSS a JavaScript. Použití ChatGPT výrazně usnadňuje a zrychluje úvodní fázi vývoje webové aplikace díky generování strukturovaného kódu, který je v souladu webovými standardy. Vývoj s pomocí ChatGPT však vyžaduje následnou kontrolu a ruční úpravy v prostředí Visual Studio Code, protože automaticky generovaný kód může obsahovat menší syntaktické nebo logické chyby, případně nemusí být vždy plně optimalizovaný pro konkrétní použití.

## **WordPress**

Z hlediska CMS platform, které se nesoustředí na přímou práci s kódem stojí za opětovné zmínění alternativy jako Bubble a Loyable, které se náročností svého workflow přiblížují spíše právě CMS řešením. V tomto případě byl ale opět zvolen jejich konkurent, protože původně jsem vytvořil webové řešení pro firmu Pila Zachotín s.r.o. pomocí nástroje WordPress.

WordPress umožňuje rychle a jednoduše vytvářet a spravovat webové stránky bez nutnosti hlubších programátorských znalostí, zejména díky množství dostupných témat a pluginů. Pro účely praktického porovnání jsem využil nejnovější stabilní verzi systému WordPress dostupnou v době psaní této práce, verzi 6.5.2. WordPress přinesl zásadní úsporu času při implementaci webových stránek, avšak flexibilita a možnosti individualizovaných řešení jsou zde oproti ručnímu kódování značně omezené.

## **Sora**

Pro generování vizuálního obsahu jsem využil AI nástroj Sora, který patří mezi novější generátory obrázků založené na hlubokém učení. Sora umožňuje na základě detailního textového popisu vytvářet vizuálně atraktivní grafické prvky, které jsou vhodné pro použití v digitálním prostředí. Hlavní výhodou je rychlosť tvorby a možnost generovat unikátní vizuální obsah přesně dle specifikace.

## **Google PageSpeed Insights**

Nakonec byla výkonnost a optimalizace webových aplikací hodnocena pomocí dvou standardních analytických nástrojů, Google PageSpeed Insights. Tyto nástroje poskytují detailní analýzu výkonu webových stránek, včetně rychlosti načítání, optimalizace obsahu a přístupnosti.

## **Canva**

Nástroj pro grafické úpravy a tvorbu jednoduchých vizuálů, byl využit například k doladění barevných filtrů nebo ořezávání generovaných a získaných obrázků. Následně byl také použit k vytvoření původního firemního loga.

## **MapChart**

Webový nástroj pro tvorbu tematických map, použitý k vygenerování grafické mapy exportních destinací firmy. Nabízí možnost ručního označení zemí a exportu mapy ve vysokém rozlišení.

## **2.2 Metodika praktické části**

Metodika praktické části této práce je postavena na porovnání tří rozdílných přístupů k vývoji webových aplikací: tradičního ručního kódování (HTML, CSS, JavaScript), generování kódu s využitím umělé inteligence a tvorby webu pomocí CMS platformy

WordPress. Cílem není vytvářet tři identické weby, ale tři funkčně rovnocenná řešení, která splňují stejný účel a požadavky kladené na výsledný produkt z pohledu uživatele.

Výchozím bodem experimentu byl web vytvořený pomocí systému WordPress, který představuje reálné řešení používané zákazníkem. Na jeho základě byla vytvořena dvě další alternativní řešení – jedno s využitím generativní umělé inteligence ChatGPT, druhé klasickým způsobem bez použití AI nástrojů.

Pro porovnání zvolených přístupů k vývoji byla využita metoda komparace, pomocí které byly analyzovány podobnosti a rozdíly mezi jednotlivými řešeními na základě definovaného souboru kritérií. Za účelem převedení tohoto hodnocení do kvantitativní podoby a zároveň zohlednění rozdílné váhy jednotlivých kritérií, byla následně aplikována vícekriteriální rozhodovací metoda – Saatyho metoda. V rámci této metody dostalo každé kritérium přiřazen váhový koeficient, tudíž byl stanoven vzájemný význam jednotlivých kritérií. Výsledné váhy se následně v kombinaci s hodnocením získaným komparací použily k výpočtu celkového skóre každého řešení, čímž bylo dosaženo transparentního srovnání přístupů.

## **2.2.1 Zvolená kritéria pro komparaci**

Součástí metodiky je také vymezení způsobu měření kvality výstupů vytvořených třemi rozdílnými přístupy. Aby bylo možné objektivně porovnat pomocí metody komparace nejen samotný proces vývoje, ale i jeho výsledky, byla stanovena soustava hodnoticích kritérií, která se vztahují jak na technickou stránku implementace, tak na vizuální komponenty. Následující výčet specifikuje jednotlivé sledované oblasti, podle nichž budou výstupy systematicky posuzovány.

### **Kritéria pro vývoj kódu**

#### **1. Technická náročnost:**

- Toto kritérium posuzuje, jaká úroveň odborných znalostí a schopností byla nutná k vytvoření funkčního webového řešení.
- Hodnotí se schopnost samostatně implementovat strukturu, styly a funkčnost, včetně optimalizace pro různé prohlížeče a zařízení.
- Vyšší hodnocení při porovnání znamená nižší technickou náročnost a větší dostupnost pro běžné uživatele.

#### **2. Doba vývoje:**

- V rámci tohoto kritéria je sledováno, kolik hodin bylo potřeba k vytvoření kompletního webu danou metodou.
- Zohledňuje se doba tvorby kódu nebo konfigurace šablon, testování, úpravy i optimalizace.
- Vyšší hodnocení při použití metody komparace znamená rychlejší vývoj, a tedy efektivnější postup.

### 3. Kvalita výstupu:

- Hodnotí se kvalita struktury HTML a optimalizace stránky z hlediska výkonu a rychlosti načítání.
- Kombinuje objektivní metriky jako počet chyb ve validaci HTML, odchylky od doporučených standardů a měření rychlosti pomocí Google PageSpeed Insights.
- Vyšší hodnocení odpovídá webu, který vykazuje lepší technickou kvalitu, rychlosť i dodržení standardů.

### Kritéria pro získávání obrázků

#### 1. Relevance a estetika:

- Hodnocena je relevance, jakožto míra, ve které obrázek plní svou roli na webové stránce a estetika, jakožto úroveň, ve které vypadá obrázek pro zákazníkovo oko esteticky.
- Relevance byla sledována pouze u AI obrazů, estetika byla porovnána u všech obrazových dvojic.
- Vyšší hodnota při porovnání znamená, že obraz lépe vystihuje svůj obsah a je vizuálně atraktivnější.

#### 2. Kvalita výstupu:

- Cílem tohoto kritéria je najít optimum mezi kvalitou z pohledu vzhledu a kvalitou z pohledu velikosti, tak aby nebyl web zatížen příliš velkými soubory, nebo obsahem s nedostatečnou kvalitou vzhledu.
- Při hodnocení je zohledňována velikost souboru a množství pixelů v obrázku jako takovém.
- Vyšší hodnota znamená obraz s dostatečně vysokou kvalitou bez nutnosti velikosti, která by negativně ovlivňovala fungování webu.

#### 3. Právní a etický aspekt:

- Míra, ve které zasahují zvolené obrázky do právního a etického prostředí.
- Zkoumá se, zda použitý obrazový materiál neporušuje autorská práva nebo nepůsobí klamavě.
- Vyšší hodnocení při komparaci značí vyšší soulad s právními a etickými standardy.

## 2.2.2 Zvolená porovnávací metoda: Saatyho metoda

Saatyho metoda, je jedním z nejrozšířenějších nástrojů pro vícekriteriální rozhodování. Jejím hlavním principem je párové porovnání všech kritérií mezi sebou z hlediska jejich vzájemné důležitosti. Hodnotitel přiřazuje každému páru číselnou hodnotu, která vyjadřuje, nakolik je jedno kritérium důležitější než druhé, tudíž vzniká porovnávací matice, na jejímž základě se následně vypočítávají váhy jednotlivých kritérií pomocí vlastních čísel. Metoda zároveň obsahuje mechanismus kontroly konzistence rozhodování, tzv. index konzistence, který ověřuje, zda přiřazené hodnoty nejsou v logickém rozporu. Pokud je konzistence

hodnocení dostatečná, lze z výsledků spolehlivě určit, která z porovnávaných variant celkově nejlépe vyhovuje stanoveným požadavkům.

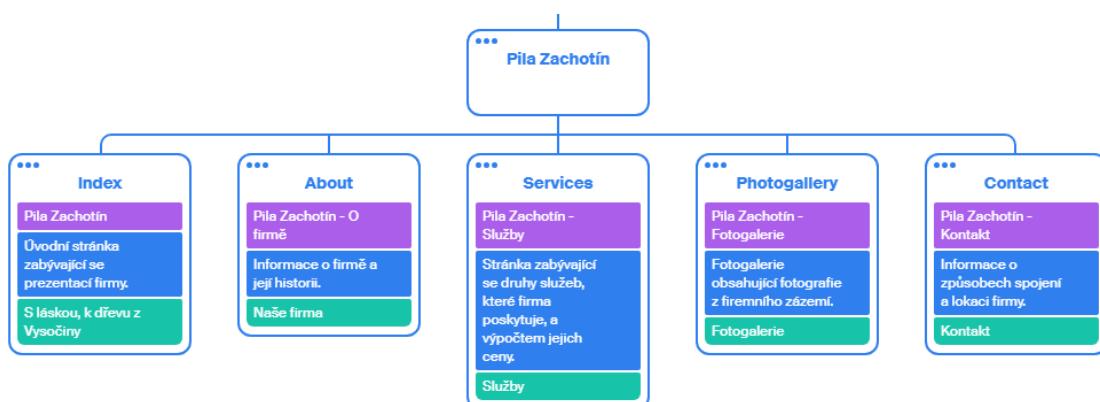
## 2.3 Vývoj kódů

### 2.3.1 Vývoj bez asistence AI

#### Html

Web vytvořený klasickým způsobem bez umělé inteligence je rozdělen do pěti hlavních stránek. Vývoj webu probíhal za užití vlastních znalostí autora, návodů na YouTube a vyhledávání na internetu.

Web byl tvořen dle následující struktury reprezentované pomocí SiteMap vytvořené za užití online nástroje pro mapování struktury stránek a její vizualizace octopus.do.



Obrázek 2 - SiteMap webu Pila Zachotín (autor)

#### Home Page (index)

Úvodní stránka slouží jako hlavní rozcestník celého webu. Na vrcholu stránky se nachází hlavička <header>, pomocí kterého se může návštěvník pohybovat po webové stránce. Uvnitř hlavičky je vložen prvek <nav>, jenž seskupuje hlavní odkazy na jednotlivé stránky webu pomocí nečíslovaného seznamu <ul>, přičemž každá jednotlivá položka menu je reprezentována značkou <li>. Logo firmy, které je vloženo jako obrázek <img>, je umístěno na levé straně navigace a plní funkci rychlého návratu na začátek stránky. Hlavíčka se ve stejném stavu nachází na všech stránkách webu.

Po navigační části následuje hero sekce, která okamžitě představuje identitu firmy. Obsahuje hlavní nadpis <h1>, doplněný třemi click to action tlačítky, která návštěvníka vedou ke klíčovým informacím o druzích poskytovaných služeb, cílových lokacích firemních produktů a lokaci firmy.

Další částí je sekce faktů o firmě, kde jsou pomocí ikon a popisků shrnutý důležité parametry firmy. V části s faktami se nachází ikony, které slouží k zjednodušení předání myšlenky a

```

<header>
  <nav class="navbar">
    <div class="navbar_container">
      
      <div class="navbar_toggle" id="mobile-menu">
        <span class="bar"></span>
        <span class="bar"></span>
        <span class="bar"></span>
        <span class="bar"></span>
      </div>
      <ul class="navbar_menu">
        <li class="navbar_item">
          <a href="index.html" class="navbar_links active" id="home-page">Domů</a>
        </li>
        <li class="navbar_item">
          <a href="about.html" class="navbar_links" id="about-page">O firmě</a>
        </li>
        <li class="navbar_item">
          <a href="services.html" class="navbar_links" id="services-page">Služby</a>
        </li>
        <li class="navbar_item">
          <a href="photogallery.html" class="navbar_links" id="photogallery-page">Fotogalerie</a>
        </li>
        <li class="navbar_item">
          <a href="contact.html" class="navbar_links" id="contact-page">Kontakt</a>
        </li>
        <li class="navbar_btn">
          <a href="tel:777-635-231" class="button" id="contact-page">Tel. +420 777 635 231</a>
        </li>
      </ul>
    </div>
  </nav>
</header>

```

Obrázek 3 - Ukázka struktury headeru vytvořená bez AI (autor)

zrychlení zachycení zaujetí návštěvníka. Stránku dále doplňuje video sekce obsahující vložené YouTube video v <iframe>, a závěrečná patička (<footer>) se sloganem a copyrightem, která se stejně jako header bude opakovat ve zbytku webové stránky.

## About us page

Stejně jako u všech dalších stránek se na vrcholu nachází hlavička. Po které následuje hero sekce, která obsahuje nadpis <h1> a podnadpis v podobě odstavce <p>. Tato sekce stručně informuje uživatele, o čem stránka je.

Hlavní část stránky tvoří sekce o prostředí firmy. V této sekci je vlevo umístěna fotografie týmu firmy v elementu <img>, zatímco vpravo je umístěn textový blok s nadpisem <h2>, podnadpisem <h4> a dvěma odstavci <p>. Cílem této kombinace je návštěvníka nejen informovat o historii firmy, ale také mu vizuálně přiblížit její prostředí a hodnoty.

Druhou klíčovou částí stránky je týmová galerie. Tato galerie představuje klíčové zaměstnance firmy. Každý člen týmu je zobrazen v samostatném bloku <div>, kde je vložen obrázek <img>, jméno v nadpisu <h3>, pracovní pozice v <span> a krátký popis v odstavci <p>. Díky tomu si návštěvník může rychle vytvořit představu o lidech stojících za firmou.

Stránku uzavírá exportní sekce, kde je vlevo textové shrnutí cílů exportu a vpravo ilustrační mapa v podobě obrázku <img>. Tato část návštěvníkovi ukazuje, do jakých zemí firma své produkty vyváží, a posiluje tak vnímání její globální působnosti.

## **Photo Gallery page**

Stránka fotogalerie slouží k vizuální prezentaci firemního zázemí a technologických zařízení. Po tradiční hlavičce <header>, která zajišťuje návaznost struktury webu, následuje hero sekce s hlavním nadpisem <h1> a stručným podtitulem v odstavci <p>. Tato úvodní sekce je strukturována stejně jako u všech ostatních stránek kromě home page.

Hlavní obsah tvoří jednotlivé galerie rozdělené do tematických bloků. Každá galerie obsahuje vlastní název v nadpisu <h3>, krátký úvodní text v <p> a několik fotografií. Fotografie jsou vloženy pomocí <img>, doplněné názvem zařízení <h4> a krátkým popisem v odstavci <p>. Galerie jsou tematicky rozdělené na expedici výrobků, vozový park, sušení dřeva a dřevozpracující techniku. Tato struktura umožňuje návštěvníkovi snadno procházet obrázky a získat informace o vybavení firmy a jejích výrobních procesech.

## **Contact page**

Stránka Kontakt je určena k tomu, aby návštěvníkům poskytla jednoduchý způsob, jak se spojit s firmou nebo ji osobně navštívit. Stejně jako u předchozích stránek začíná hlavičkou, je zakončena patičkou a nachází se zde také stejná hero sekce.

Hlavní část stránky tvoří kontaktní sekce rozdělená do dvou bloků. První blok obsahuje základní kontaktní informace jako název firmy, adresu, telefonní číslo a e-mail, strukturované v elementech <address>, <tel> a odkazech <a>. Druhý blok tvoří vložená mapa přes <iframe>, která ukazuje přesnou polohu firmy.

## **Services page**

Hlavní část stránky tvoří seznam služeb, který je strukturován pomocí nečíslovaného seznamu <ul>, kde každá položka <li> představuje jednu konkrétní službu. Výběr služeb je připraven tak, aby návštěvník jedním kliknutím získal více informací a možnost výpočtu orientační ceny. Pod seznamem služeb je umístěn kalkulátor. Kalkulátor využívá vstupní pole <input> a výběrové prvky <select> pro zadání parametrů jako je druh dřeva, třída kvality a objem materiálu. Výstup kalkulace je zobrazen dynamicky v textovém prvku <div>, aby návštěvník okamžitě viděl odhad ceny bez nutnosti reloadu stránky.

## **CSS**

Pro základní barevné schéma byla použita kombinace přírodní hnědé barvy - RGBA(99, 81, 78, 0.8), bílé a černé barvy. Hnědá barva se uplatňuje zejména v hlavičce, na pozadí tlačítek a v dalších akcentových prvcích, kde evokuje materiál spojený s činností firmy. Bílá barva byla využita pro pozadí hlavního obsahu a textových bloků, čímž se dosáhlo dostatečného kontrastu a zvýšila se čitelnost. Černá barva slouží k poukázaní na kontrast barvy bílé.

Typografie byla sjednocena použitím písma Kumbh Sans, které bylo vybráno pro svou jednoduchost a dobrou čitelnost na různých zařízeních. Hierarchie textu je stanovena použitím odlišných velikostí a vah písem pro nadpisy a běžný text. Struktura rozvržení jednotlivých sekcí je založena převážně na technice Flexbox. Tato technika je využita například pro horizontální zarovnání navigačního menu a vycentrování prvků v hero sekcích.

Obrázky na webu jsou využívány zejména v hero sekcích a v galeriích. V hero sekcích je aplikován velkoformátový obrázek dřevěné textury s použitím lineárního překryvu, což umožňuje současné zachování pozadí i čitelnosti textu. V galerijních sekcích a v části o firmě byly obrázky optimalizovány pro rychlejší načítání stránky, přičemž byla zachována dostatečná kvalita vizuální prezentace.

Odsazení, mezery mezi prvky a zarovnání textů byly řešeny s ohledem na konzistenci napříč jednotlivými sekcemi. Responzivní úpravy byly realizovány pomocí media queries, především pro obrazovky se šířkou pod 768 px, kde bylo upraveno například rozvržení menu a velikostí textu.

```
/* Hero sekce */
.hero {
    background-image: linear-gradient(rgba(0,0,0,0.1), rgba(0,0,0,0.1)), url(images/wooden-hearth.jpg);
    background-position: center;
    background-size: cover;
    position: relative;
    padding: 250px 0 200px 0;
    margin-top: 0;
}
.hero__container {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    max-width: 1200px;
    margin: 0 auto;
    height: 90%;
    text-align: center;
    position: relative;
    padding: 30px;
}
.hero__heading {
    font-size: 100px;
    margin-bottom: 24px;
    color: #fff;
}
.hero__btns {
    display: flex;
    justify-content: center;
    gap: 40px;
    margin-top: 30px;
    flex-wrap: wrap;
}
.hero__btn {
    background-color: rgba(99, 81, 78, 0.8);
    color: #ffffff;
    text-decoration: none;
    padding: 12px 20px;
    margin-top: 50px;
    font-size: 1.5rem;
    border-radius: 3px;
    transition: all 0.3s ease;
}
```

Obrázek 4 - Ukázka CSS stylování sekce hero vytvořené bez AI (autor)



Obrázek 5 - Vzhled domovské stránky vytvořená bez AI (autor)

## JS

Javascript webu lze rozdělit na 2 části. Část zabývající se interaktivitou a responsibilitou a část zabývající se kalkulátorem ceny služeb. Proto je tato část práce rozdělena do dvou podkapitol – navigace stránky a kalkulátor ceny služeb.

### Navigace stránky

První oblast využití JavaScriptu se zaměřuje na ovládání navigačního menu a podporu hladkého návratu na začátek stránky. Interaktivní mobilní menu bylo řešeno přidáváním a odebíráním tříd pomocí metody `classList.toggle()`. Kliknutím na ikonku mobilního menu (`#mobile-menu`) se menu zobrazí nebo skryje, čímž je zajištěna použitelnost navigace na menších zařízeních.

Součástí navigace je také funkce umožňující návštěvníkovi plynule se vrátit na začátek stránky po kliknutí na logo (`#navbar__logo`). Skript využívá metodu `scrollTo` s nastaveným behavior smooth, což přispívá k lepšímu uživatelskému zážitku a orientaci na stránce.

```
/* Hamburger menu pro mobilní zařízení */
const menu = document.querySelector('#mobile-menu');
const menuLinks = document.querySelector('.navbar__menu');
const navLogo = document.querySelector('#navbar__logo');

const mobileMenu = () => {
    menu.classList.toggle('is-active');
    menuLinks.classList.toggle('active');
};

/* Scroll na začátek stránky při kliknutí na logo */
menu.addEventListener('click', mobileMenu);
document.getElementById('navbar__logo').addEventListener('click', function (e) {
    e.preventDefault();
    window.scrollTo({ top: 0, behavior: 'smooth' });
});
```

Obrázek 6 - JS logika pro navigaci stránky vytvořená bez AI (autor)

## Kalkulátor ceny služeb

Druhá část JavaScriptu je věnována dynamickému kalkulátoru služeb, který umožňuje návštěvníkovi orientačně vypočítat cenu požadované služby na základě zadaných parametrů. Kalkulátor je propojen s HTML formulářem obsahujícím výběrové seznamy pro druh dřeva a kvalitu řeziva a vstupní pole pro objem materiálu.

Data o jednotlivých službách jsou uložena v objektu services, který obsahuje název, popis a základní cenu každé služby. Násobící koeficienty pro typ dřeva a jeho kvalitu jsou definovány v samostatném objektu woodMultipliers. Výpočet je proveden v několika krocích. Nejprve je na základě vybrané služby, druhu dřeva a kvality vypočtena cena za metr krychlový. Pokud je zadán objem materiálu, celková cena je následně určena vynásobením této jednotkové ceny a objemu.

Skript reaguje na kliknutí na jednotlivé služby v seznamu, změny ve výběrových prvcích i na úpravy objemu materiálu. Výsledky jsou ihned dynamicky zobrazeny na stránce, aniž by bylo nutné ji znova načítat.

```
/* Funkce updatující celkovou cenu v kalkulátoru */
function updateTotal() {
    const volume = parseFloat(volumeInput.value);
    const woodType = woodTypeSel.value;
    const woodGrade = woodGradeSel.value;
    const multiplier = woodMultipliers[woodType] * woodMultipliers[woodGrade] || 1;
    if (service) {
        const adjustedPrice = service.price * multiplier;
        price.textContent = adjustedPrice.toLocaleString("cs-CZ");
        if (!isNaN(volume)) {
            const total = volume * adjustedPrice;
            totalPrice.textContent = total.toLocaleString("cs-CZ", {
                minimumFractionDigits: 2,
                maximumFractionDigits: 2
            });
        } else {
            totalPrice.textContent = "0";
        }
    }
}
```

Obrázek 7 - JS logika kalkulátoru funkce update vytvořená bez AI (autor)

### 2.3.2 CMS vývoj

Tento přístup k vývoji webu je reprezentovaný tvorbou webu prostřednictvím systému WordPress, nevyužívá přímou práci s HTML, CSS a JavaScriptovým kódem. Z tohoto důvodu není vhodné provádět přímé porovnání na úrovni zdrojového kódu. Analýza této části se proto zaměřuje na strukturu stránek, vzhled, způsob stylování a základní interaktivitu dosaženou prostřednictvím nástrojů NoCode platformy.

WordPress byl nasazen na hostingovou službu a základně nakonfigurován. Po instalaci systému byla aktivována šablona Sonoran, která odpovídala požadavkům na jednoduchý, přehledný a responzivní firemní web. Navigační struktura byla nastavena vytvořením hlavního menu obsahujícího odkazy na jednotlivé stránky. Menu bylo propojeno se všemi hlavními částmi webu a bylo vloženo do hlavičky, aby bylo dostupné z každé stránky.

Na Home page byla vložena hero sekce se sloganem „S láskou k dřevu z Vysočiny“, podpořená pozadím s dřevěným motivem. Pod hlavním sdělením byla přidána tři call to action<sup>5</sup> tlačítka vedoucí na jednotlivé sekce webu. Struktura této stránky byla doplněna o sekci s klíčovými fakty o firmě a vloženým propagačním videem.

Stránka O firmě byla sestavena pomocí obsahových bloků obsahujících texty o historii firmy a fotografie z provozu. Textové bloky byly uspořádány tak, aby byly snadno čitelné a vizuálně přitažlivé. Stránka Služby byla vytvořena vložením jednotlivých obsahových bloků pro každou nabízenou službu. U každé služby byl uveden název, stručný popis a v některých

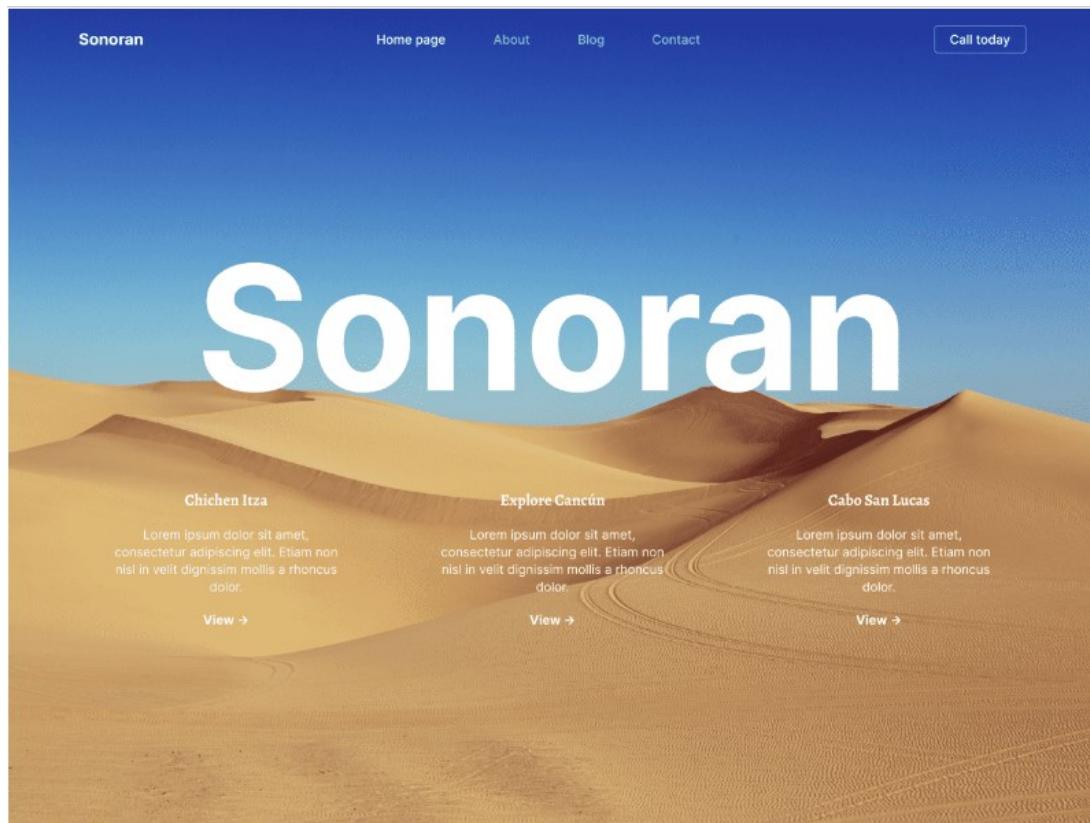
---

<sup>5</sup> Nadále CTA

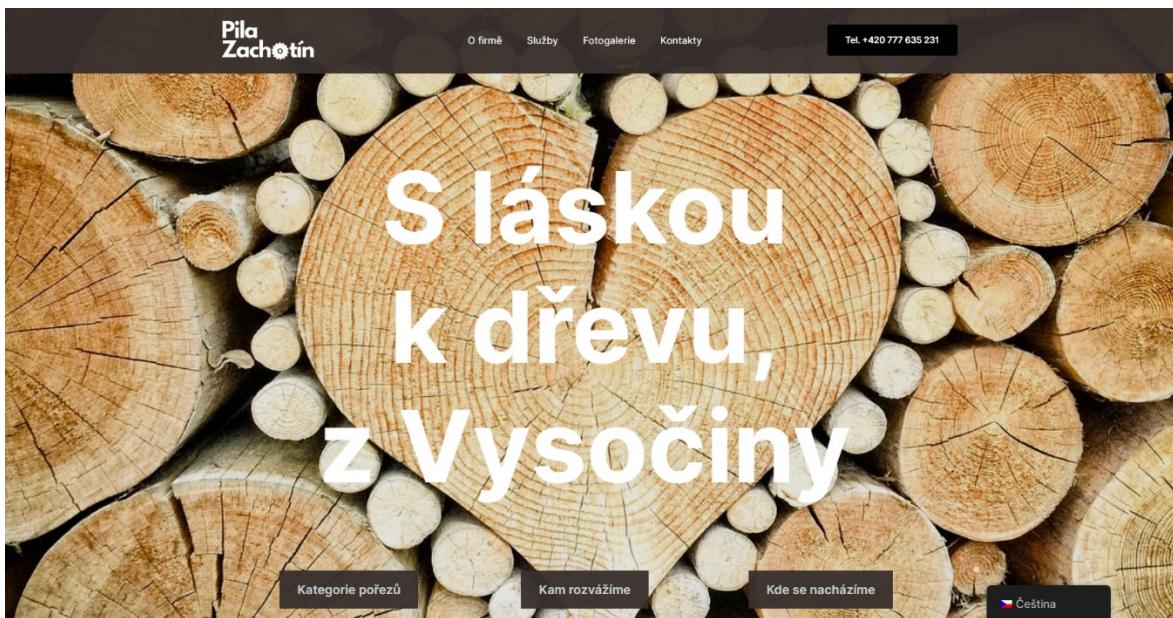
případech i orientační cena. Bloky byly seřazeny za sebou pro dosažení přehledného uspořádání nabídky. Fotogalerie byla sestavena nahráním fotografií do gridového rozvržení. Obrázky byly organizovány do tematických skupin podle typu činnosti – výroba, expedice, skladování.

Stránka Kontakt byla vytvořena vložením bloku s textovými kontaktními údaji a následně vloženou mapou Google Maps, která umožňuje návštěvníkům snadno najít lokaci firmy. Kontaktní formulář na webu nebyl implementován, komunikace je směrována na e-mailovou adresu a telefonní číslo.

Responzivita stránek byla zajištěna automatickými funkcemi šablony Sonoran. Zobrazení na mobilních zařízeních bylo zkонтrolováno a byly provedeny drobné úpravy rozložení prvků pro zajištění správného fungování na menších obrazovkách.



Obrázek 8 - Vzhled šablony využívané v nástroji WordPress (autor)



Obrázek 9 - Vzhled úvodní stránky při vývoji s CMS (autor)

### 2.3.3 AI asistovaný vývoj

Posledním druhem vývoje je vývoj s využitím AI. V tomto přístupu byl web vyvíjen za použití promptů jakožto nástroje pro komunikaci s chatbotem Chat GPT 4.5. Proto bude tato část práce zaměřena na prompty jako takové a jejich output z hlediska vzhledu celé stránky, a ne z hlediska struktury kódu, jako tomu bylo v části první.

#### Inicializace

Použitý prompt: *You are creating a web page for a sawmill which is located in the Czech Republic... The page will have 5 subpages which you can navigate through using the navbar (Domů, O firmě, Služby, Fotogalerie, Kontakt) ... We will start by creating the index page (Domů), specifically the header and the footer... Do not program yet I will specify what I want from each component, just answer me whether you are capable of doing so.*

Tento úvodní prompt sehrál zásadní roli v zahájení generativní tvorby webových stránek Pily Zachotín. Nešlo pouze o informativní vstup, ale o iniciační specifikaci, která vygenerovala několik klíčových rozhodnutí a strukturálních konstant celého projektu:

- Ujasnění účelu: webová prezentace pily se sídlem v České republice.
- Stanovení základní navigační struktury webu.
- Určení výchozí stránky Domů jako index.html.
- Deklarace metodiky vývoje: postupný návrh po komponentách s možností uživatelského zásahu mezi jednotlivými kroky.

Tento prompt nesloužil pouze jako formální začátek, ale jako architektonický základ celého vývoje. Jeho role odpovídá tzv. Design Intent Declaration nebo Initial Scope Setup, tedy bodu, ve kterém se definuje rozsah, účel a metodologie projektu.

## Home page

**Header** – použitý prompt: *Ok lets start by creating the header. I want there to be a logo on the left side which says Pila Zachotín (Instead of the (o) in the logo I want there to be a saw, also the logo should be stacked to the left and the two words should be below each other. When you click on the logo you should scroll up to the top of the page (use js)*

*Then I want there to be a navbar with the 5 subpages and a button with a telephone number on the right side.*

*The header should be made in a way so its as best as possible (your life depends on it). It should be responsive and good looking. When the resolution is lowered it should be replaced with a button which opens a menu which will contain the navbar (written with js – use folder app.js)*

*I want it to be brown but partly seethrough because in the future I want the hero parts of the pages to be behind the header. Also keep the header visible at all times when you scroll down.*

*Take a deep breath and start programming while keeping in mind all the previously mentioned requirements.*

Tento prompt detailně definuje jeden z nejzásadnějších prvků celého webu – hlavičku, která zajišťuje:

- vizuální identitu – logo s pilou,
- navigační strukturu – odkazy na podstránky,
- interaktivitu a přístupnost – telefonní tlačítko, scrollToTop,
- responzivitu a adaptivitu – hamburger menu.

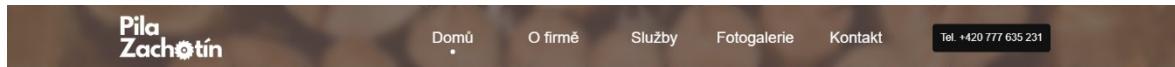
Při implementaci tohoto promptu nastal problém, protože generativní model původně nevytvořil logo jako samostatný obrázek, ale jako součást HTML kódu ve formě SVG prvku, což vedlo k nízké grafické kvalitě a problémům se zarovnáním písmen a pilového kotouče. Z tohoto důvodu se autor rozhodl logo nevytvářet programově, ale použít generativní model pro tvorbu obrázků, ve kterém nechal logo vygenerovat jako samostatný obrázek a poté jej vložil do hlavičky webu jako klasický <img> prvek.



Obrázek 10 - Vzhled headeru při vývoji s AI (autor)

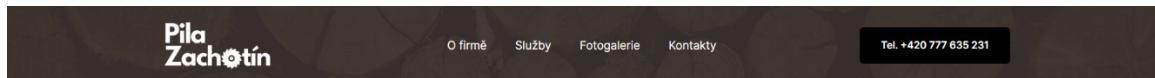
V porovnání hlavičky vytvořené pomocí AI a hlavičky sestavené bez jeho použití můžeme hned z počátku vymezit hlavní rozdíl ve vzhledu firemního loga. Touto části headeru, se ale budeme zabývat až v následující části o vygenerovaných obrázcích, tudíž zbývající rozdíly jsou hlavně:

- podtržení aktivní stránky v kontrastu s tečkou v no AI verzi,
- menší rozestupy mezi jednotlivými stránkami v navbaru,
- vzhled telefonního tlačítka – absence tel a velikost.



Obrázek 11 - Vzhled headeru při vývoji bez AI (autor)

Když porovnáme AI i no AI hlavičku s headerem vytvořeným pomocí CMS, můžeme zaznamenat velkou podobnost právě s no AI stylováním, ale i přes to zde lze nalézt rozdíl, například v centrování navbaru a v barvě pozadí. Z pohledu hlavičky vytvořené pomocí AI zde přetrvává rozdílné logo a výše uvedené rozdíly.

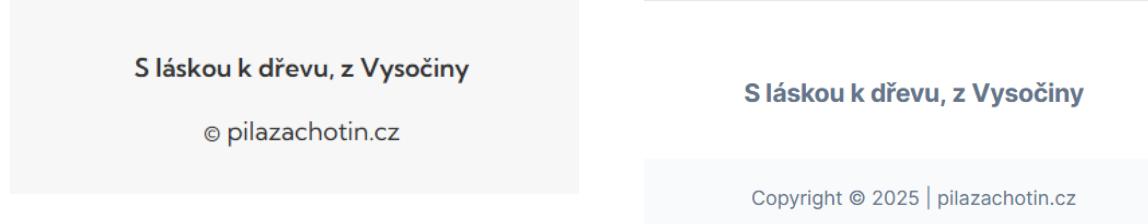


Obrázek 12 - Vzhled headeru při vývoji s CMS (autor)

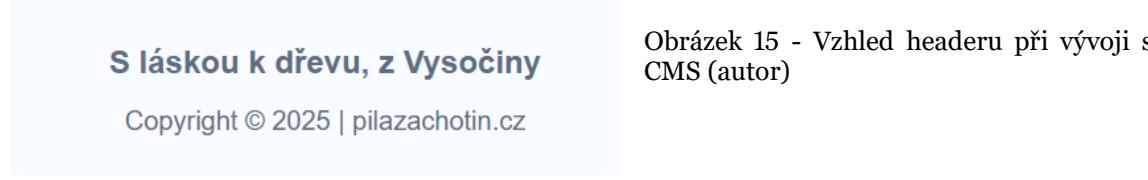
**Footer** – použitý prompt: *Now create a footer, there will only be a simple line (S láskou k dřevu, z Vysočiny) and a copyright to pilazachotin.cz stacked above each other (clickable link to the website). The footer should be separated from the rest of the page with a grey line and a slight color change in comparison to normal white background.*

Nahoře uvedený prompt se zabývá popisem stavby patičky stránky. Ve footeru se nachází pouze jednoduché moto firmy s copyrightem odkazujícím na firmu a následující popis jejich zobrazení již zmíněném footeru.

Při porovnání patiček všech tří vytvořených variant není vidno mnoha rozdílů. Největší rozdíly jsou opět viditelné převážně při porovnání varianty s využitím AI se zbylými dvěma, jelikož je v ní znění footeru pozměněné, na druhou stranu je v tomto případě spacing verze s využitím AI podobnější než AI spacingu a výrazněji odlišný od CMS.



Obrázek 13 - Vzhled headeru při vývoji s AI (autor)



Obrázek 14 - Vzhled headeru při vývoji bez AI (autor)

**Hero** – použitý prompt: *Now we will add a big hero part to the page.*

*Hero will have a big hearth made out of wooden logs with logs all around it on the background (I want you to generate it as an image).*

*It will contain a simple h1 saying S láskou k dřevu, z Vysočiny.*

*Below the h1 there will be 3 buttons saying Co vyrábíme, Kam rozvážíme and Kde se nacházíme. These buttons will be later linked to direct locations on other pages using ids of the elements using href. I want the background of buttons to be the same as in the header previously.*

*Keep in mind the rules of the responsivity of the whole webpage and take a deep breath and start working.*

V případě tohoto promptu je jeho cílem vytvoření hero části home page, která bude sloužit k zaujetí zákazník a představení key pointů pomocí CTA tlačítek.

Do středu sekce byl umístěn nadpis s textem „S láskou k dřevu, z Vysočiny“, který tvoří hlavní sdělení firmy. Pod nadpisem byly umístěny tři tlačítka – „Co vyrábíme“, „Kam rozvážíme“ a „Kde se nacházíme“ – která slouží jako interní odkazy na konkrétní sekce webu prostřednictvím budoucích id atributů. Prompt výslově požadoval, aby tlačítka barevně odpovídala předchozí hlavičce, což vedlo AI k jejich zbarvení do poloprůhledné hnědé.

Po vytvoření hero sekce opět, stejně jako u sekce header, vznikl problém s vygenerovaným obrázkem, který neodpovídal autorově představě, a proto byl následně přegenerován pomocí specializovaného asistenta.

V případě porovnání AI řešení s dvěma zbývajícími můžeme vidět, že se hero sekce ze strany vzhledu textu a tlačítek moc neliší. Hlavním rozdílem je velikost textu, který je u CMS řešení výrazně větší, ve zbytku řešení ale nejsou znatelné odchylky. Klíčovým rozdílem je tudíž obrázek na pozadí, který byl v AI řešení generován na rozdíl od zbylých dvou řešení, využívajících obrázek z databanky.



Obrázek 16 - Vzhled hero při vývoji s AI (autor)



Obrázek 17 - Vzhled hero při vývoji bez AI (autor)



Obrázek 18 - Vzhled hero při vývoji s CMS (autor)

**Sekce ikon** – použitý prompt: *Now add a part below the hero which will contain icons and text next to them in a small row.*

*There will be 4 icons. Each one will have 2 rows of text on the right side. The 2 rows should be the same height as the icon. Top row will have blue color. Bottom row black. The icons will also be black. Background kept white same as the whole page. The 4 icons will be next to each other in a row - keep responsibility in mind.*

*Icon 1: (icon - wood cutting saw, top row – 1000m<sup>3</sup>, bottom row - Měsíční kapacita pořezu)*

*Icon 2: (icon - broadleaved tree, top row -60%, bottom row – Pořez listnatých)*

*Icon 3: (icon - coniferous tree, top row -40%, bottom row - Pořez jehličnatých)*

*Icon 4: (icon - a ruler, top row -Tolerance +/- 1mm, bottom row - Přesnost pořezu)*

Tento prompt definoval vznik nové sekce umístěné bezprostředně pod hero částí úvodní stránky, jejímž cílem bylo stručně a vizuálně atraktivně prezentovat čtyři klíčové výrobní parametry firmy. Zadání obsahovalo jasnou strukturu: čtyři ikonové bloky umístěné v jedné řadě, přičemž každý z nich měl po pravé straně dvě textové řádky – horní zvýrazněnou modře a spodní černě. Ikony měly být jednoduché, černé, tematicky odpovídající zadání: pila, listnatý strom, jehličnan a pravítko.

Na základě těchto požadavků vygeneroval asistent ikony, avšak jejich verze byla nedostačující – velké množství chyb a nepřesný vzhled. Z tohoto důvodu muselo být opět přistoupeno k variantě přegenerování ikon v jiném asistentu.

V tomto případě opět vznikla neočekávaná situace toho, že sekce vytvořená pomocí AI byla podobnější řešení CMS ve více ohledech než sekce bez využití AI, která brala CMS jako svůj vzor. Níže můžeme vidět, že jako jediná je sekce bez AI vycentrována do středu na rozdíl od čehož je sekce z CMS a AI vycentrována vlevo. Ostatní rozdíly, kromě vzhledu ikon nejsou znatelné.



Obrázek 19 – Vzhled sekce ikon při vývoji s AI (autor)



Obrázek 20 – Vzhled sekce ikon při vývoji bez AI (autor)



Obrázek 21 - Vzhled sekce ikon při vývoji s CMS (autor)

**Sekce videa** – použitý prompt: *Now add the last part of the domů page (index) this part will contain a link to a YouTube video with a title Natočili o nás. Keep responsivity in mind.*

V posledním zadání pro část domů byl zvolen velice jednoduchý prompt pro velice jednoduchý požadavek, a to vytvoření sekce s videem. V této sekci vznikly absolutně minimální odchylinky, a proto nevidím důvod porovnávání jejího vzhledu.

## Services page

**Hero** – použitý prompt: *Ok now we will create a new page the Služby page.*

*The header and footer will be kept the same, but the hero section will be changed a little.*

*The background image of the hero section will be kept but the section itself will have lower height so the image is to be cropped (dont alter the image just use css). Also there will be an h1 saying Fotogalerie, alligned to the left but with some margin and a one sentence*

*text below the h1 saying Foto vhled do naší firmy. The text color and all will be same as the hero section in index, dont add any buttons tho.*

Stránka Služby vznikla na základě promptu, jehož cílem bylo vytvořit novou podstránku s odlišnou, ale vizuálně obdobnou hero sekcí. Zadání jasně stanovovalo, že se má zachovat stávající struktura hlavičky a patičky, aby byla zajištěna konzistence napříč celým webem. Pozornost se soustředila především na změnu výšky hero sekce – oproti úvodní stránce měla být výškově kompaktnější a obrazové pozadí mělo být pouze oříznuto pomocí kaskádových stylů, bez zásahu do samotného obrazového souboru.

Textový obsah hero bloku byl zformulován tak, aby ladil s vizuálním stylem zvoleným již dříve. Nadpis umístěný vlevo doplněný jednou vysvětlující větou měl za úkol definovat charakter stránky a zároveň nenarušovat minimalistické pojetí rozvržení. Styl písma, barvy a pozicování byly převzaty z úvodní stránky, čímž byla podtržena vizuální jednota celého rozhraní.

Při porovnávání vzhledů jednotlivých řešení vzniklo značné množství rozdílů. U řešení s využitím AI je zřejmé pozicování textu blíže k levé straně a vršku stránky. U no AI řešení vidíme obdobnou délku a rozsah, ale pozicování dále od levého okraje a vršku stránky. CMS řešení na rozdíl od dvou předchozích postrádá popisek a v porovnání s no AI řešením je barva jeho pozadí výrazně světlejší.



Obrázek 22 - Vzhled hero2 při vývoji s AI (autor)



Obrázek 23 - Vzhled hero2 při vývoji bez AI (autor)



Obrázek 24 - Vzhled hero2 při vývoji s CMS (autor)

**Kalkulátor ceny služeb** – použitý prompt: *Create a calculator for the Služby page which will calculate pricing of specific services (pořez stavebního řeziva, pořez truhlářského*

*řeziva, výroba pražců, výroba podlahových lamel, výroba palet a obalů, sušení řeziva a impregnace řeziva) based on the type of wood (smrk, modrin, borovice,dub,buk) and wood grade (a – highest, b, c, d, e).*

*You should be able to input the amount of wood in m3 choose a service and than select what grade and type you want to use.*

*Select the pricing of the services and multipliers based on the file i provided. Make the calculator look professional it is going to be the center of this page. Keep responsivity in mind.*

Hlavním požadavkem v promptu bylo vytvoření kalkulátoru cen služeb. Cílem tohoto kalkulátoru bylo určení ceny specifické služby na základě předem zvolených parametrů. Mezi kategorie, které mění cenu služeb patří druh dřeva a třída dřeva z pohledu kvality. Tento kalkulátor má sloužit k zjednodušení přístupu zákazníka k informacím o službách v porovnání s původním webem.

CMS verze webu kalkulátor jako takový neobsahuje, jelikož si firma nepřála operovat ve svém CMS s komplikovanými pluginy, které i přes to nedokázaly naplnit jejich požadavky na kalkulátor, tudíž byl vytvořen pouze u AI a no AI řešení. U CMS řešení nahrazuje kalkulátor jednoduchý seznam poskytovaných služeb.

AI verze kalkulačky je postavena jako klasický webový formulář. Uživatel vyplní objem, vybere službu, typ dřeva a třídu kvality a výpočet je spuštěn odesláním formuláře. Hodnoty jsou strukturovány do tří objektů: basePrice, typeMult, gradeMult, a výsledná cena je vypočítána jedním výrazem ve funkci submit. Výstup je zobrazen jako jedna finální orientační cena ve formátu měny, bez zobrazení jednotlivých dílčích cen nebo popisů služeb.

**Kalkulátor ceny služeb**

Objem řeziva (m<sup>3</sup>)

Služba

Pořez stavebního řeziva

Druh dřeva

Smrk

Třída kvality

A – nejvyšší

Spočítat cenu

Obrázek 25 - Vzhled kalkulátoru při vývoji s AI (autor)

Pila Zachotin

## Cenový kalkulátor služeb

Vyberte si službu a pár jednoduchými kliknutími se dostanete na odhadovanou cenu služby.

[Pořez stavebního řeziva](#)

[Pořez truhlářského řeziva](#)

[Výroba pražců](#)

[Výroba podlahových lamel](#)

[Výroba palet a obalů](#)

[Sušení řeziva](#)

[Impregnace řeziva](#)

Podrobnosti služby

### Pořez stavebního řeziva

Zajišťujeme pořez stavebního řeziva dle specifikaci zákazníka.

Cena za m<sup>3</sup>: 7 000 Kč

Typ dřeva:

Smrk

Kvalita dřeva:

A (nejvyšší)

Zadejte množství (m<sup>3</sup>):

Např. 2.5

Odhadovaná cena: 0 Kč

Obrázek 26 - Vzhled sekce 1 kalkulátoru při vývoji bez AI (autor)

Obrázek 27 - Vzhled sekce 2 kalkulátoru při vývoji bez AI (autor)

Tento přístup je technicky efektivní a z hlediska čitelnosti velmi kompaktní. Kód neřeší pokročilou interaktivitu, ale zaměřuje se na rychlý výpočet a jednoznačný výstup. Výhodou je jednoduchost a modularita, nevýhodou pak omezená interaktivní zpětná vazba pro uživatele. V porovnání s tímto řešením má no AI verze výrazně složitější strukturu, která ale dovoluje uživateli zisk většího množství informací – kromě výpočtu ceny zobrazuje i popis zvolené služby a odděleně prezentuje jednotkovou i celkovou cenu. Vybrané řešení tedy vyžaduje více kódu a logiky, ale poskytuje uživateli detailnější zpětnou vazbu.

```
/* ----- PRICE CALCULATOR ----- */
document.addEventListener('DOMContentLoaded', () => {
  const form = document.getElementById('calc-form');
  const out = document.getElementById('result');

  const basePrice = {           // CZK / m3
    rezivostaveb: 2500,
    rezivotruhlar: 3000,
    prazce: 3200,
    lamely: 3500,
    palety: 2800,
    suseni: 800,
    impregnace: 600
  };

  const typeMult = { smrk:1.00, modrin:1.10, borovice:1.05, dub:1.40, buk:1.30 };
  const gradeMult= { a:1.20, b:1.10, c:1.00, d:0.90, e:0.80 };

  form.addEventListener('submit', e => {
    e.preventDefault();

    const m3      = parseFloat(form.volume.value) || 0;
    const service = form.service.value;
    const wood    = form.woodType.value;
    const grade   = form.grade.value;

    const price = m3 * basePrice[service] * typeMult[wood] * gradeMult[grade];

    out.textContent =
      isFinite(price) && price > 0
        ? `Orientační cena: ${price.toLocaleString('cs-CZ', { style:'currency', currency:'CZK', maximumFractionDigits:0 })}`
        : 'Zadejte prosím platné údaje.';
  });
});
```

Obrázek 28 - JS logika kalkulátoru vytvořeného s AI (autor)

Popis tvorby Home page a Services page pokrývá většinu klíčových komponent, interaktivních prvků i vizuálních struktur, které definují funkční a estetický rámec celého webu. Tyto dvě stránky sloužily jako základní modely pro implementaci jak statického, tak dynamického obsahu, a nastavily systém práce s prompty, jenž byl dále konzistentně uplatňován. Zbývající stránky Contact, About us a Photo Gallery sice obsahují specifické textové a obrazové bloky, nicméně jejich implementace vychází z totožného přístupu k návrhu pomocí promptů: opakované využití existujících komponent, úpravy jejich parametrů dle zadání a respektování vizuální funkční konzistence napříč celým webem. Z toho důvodu nepokládá autor za důležité je zmiňovat v této části práce.

### 2.3.4 Souhrn

#### Technická náročnost

Ruční vývoj webu pomocí HTML, CSS a JavaScriptu byl technicky nejnáročnější. Každý prvek musel být ručně naprogramován, včetně zajištění responzivity, funkčnosti interaktivních prvků a optimalizace výkonu. Bylo nutné mít znalosti standardů webového vývoje a schopnost řešit kompatibilitu napříč různými prohlížeči a zařízeními. Tento postup vyžadoval jak znalost syntaxe, tak schopnost architektonického návrhu webu.

Použití generativní umělé inteligence snižovalo technickou náročnost vývoje. Většina kódu byla vygenerována automaticky na základě textových promptů, přičemž hlavní technickou výzvou byla kontrola kvality výstupů a jejich případná ruční úprava. I přesto bylo nutné rozumět generovanému kódu, aby bylo možné provádět opravy a optimalizace, zejména v oblastech jako je struktura dokumentu, responzivita a SEO.

Technická náročnost v případě CMS řešení byla nejnižší. Většina funkcionality byla zajištěna prostřednictvím předdefinovaných bloků a šablon, které bylo možné upravovat pomocí grafického rozhraní. Nastavování vzhledu, navigace a obsahu nevyžadovalo znalost programování. Technická výzva spočívala hlavně v orientaci v možnostech šablony Sonoran a ve schopnosti efektivně využít dostupné nástroje.

Tabulka 1 - Technická náročnost vývoje (autorské zpracování)

Způsob vývoje	Technická náročnost
Vývoj bez AI	Vysoká
Vývoj s AI	Střední
CMS vývoj	Nízká

#### Doba vývoje

Vývoj webu tradičním způsobem, tedy ručním psaním kódu v HTML, CSS a JavaScriptu, představoval největší časovou investici. Každá sekce musela být navržena, implementována a stylována samostatně, včetně testování funkčnosti a optimalizace pro různá zařízení.

Tento způsob vývoje vyžadoval největší znalosti a zkušenosti a využil široký rozsah manuální práce, což se odrazilo i na časové náročnosti, která dosáhla hodnoty 19 hodin.

Použití generativní umělé inteligence při tvorbě webu vedlo ke značné úspoře časové náročnosti. Web byl navržen prostřednictvím promptování a automatizovaného generování kódu, přičemž hlavní činnost spočívala ve formulaci požadavků, kontrole výstupů a jejich drobných úpravách. AI významně urychlila tvorbu jednotlivých sekcí a minimalizovala nutnost manuální implementace. Celková doba vývoje webu s využitím AI byla pouze 9 hodin.

Tvorba webu pomocí bez použití kódu řešení, konkrétně prostřednictvím WordPressu a šablony Sonoran, byla nejrychlejším postupem. Web byl sestavován pomocí předpřipravených bloků a vizuálního editoru, bez potřeby programování. Práce spočívala především v úpravě obsahu, rozmístění sekcí a stylování vzhledu. Tento způsob umožnil sestavení kompletního webu během 4 hodin.

Tabulka 2 - Časová náročnost vývoje (autorské zpracování)

Způsob vývoje	Časová náročnost (h)
Vývoj bez AI	19
Vývoj s AI	9
CMS vývoj	4

## Kvalita výstupu

### Kvalita HTML

Pro hodnocení kvality HTML kódu byla využita online služba W3C Markup Validation Service, která ověřuje správnost struktury a syntaxe podle standardů HTML. Výsledky ukazují rozdíly mezi jednotlivými přístupy.

Ve verzi vytvořené bez využití AI byly zaznamenány pouze jednotky chyb nebo upozornění, přičemž na většině podstránek se nevyskytla žádná chyba. Tato skutečnost vzniká z mýry kontroly nad kvalitou kódu při ručním vývoji.

Naopak stránka vytvořená s využitím AI vykazuje o něco vyšší počet chyb. Důvodem jsou obecné požadavky v promptech použitých autorem a jejich následné vyhodnocení, při kterém se AI nedrží standardů pro HTML. Navzdory tomu však počet chyb zůstává nízký a výstup je i nadále funkční.

Nejvíce chyb vykazuje CMS řešení vytvořené ve WordPressu. Každá z podstránek obsahuje 8 až 11 chyb a upozornění. Tento výsledek však není neobvyklý, WordPress generuje velké množství kódu automaticky prostřednictvím šablon a pluginů, které nejsou striktně optimalizovány pro W3C validitu. Například se zde často objevují duplicitní ID, chybějící alternativní texty u obrázků nebo zastaralé atributy.

Z metodického hlediska je tedy třeba W3C výsledky u CMS řešení interpretovat s rezervou. Web jako celek může být plně funkční, přístupný a přehledný, i přesto že nesplňuje striktní požadavky validátoru. V rámci hodnocení kvality kódu však tyto rozdíly dobře ilustrují míru technické kontroly, kterou má vývojář v jednotlivých přístupech k dispozici.

Tabulka 3 - Výsledky W3C html validace (autorské zpracování)

Způsob vývoje/Stránka	Home page	About us page	Services page	Photogallery page	Contact page
<b>Vývoj bez AI</b>	1(1)	0	0	0	1
<b>Vývoj s AI</b>	2(1)	1	4	1	1
<b>CMS vývoj</b>	11	9(1)	9(1)	9(1)	8(1)

Čísla v tabulce určují počet Error v html validaci W3C a čísla v závorce jsou počet Warning v této validaci

### Výkon a rychlosť načítania webu

Hodnocení výkonnosti webových řešení bylo provedeno pomocí nástroje Google PageSpeed Insights, který analyzuje stránky z hlediska čtyř základních oblastí: Výkon, Přístupnost, Doporučené postupy a SEO. Výkon měří rychlosť načítání a efektivitu práce s obsahem, především na mobilních zařízeních. Přístupnost hodnotí, do jaké míry je stránka použitelná i pro uživatele se specifickými potřebami. Doporučené postupy reflekují technickou správnosť a bezpečnosť webu. Posledním aspektem je SEO, které vyjadřuje optimalizaci stránky pro vyhledávače a její schopnosť být správně indexována. Všechny hodnoty jsou prezentovány jako skóre od 0 do 100.

V případě řešení vytvořeného tradičním způsobem jsou výsledky nejlepší. Hodnoty výkonu, které toto řešení vykazuje, jsou nejvyšší na všech zařízeních, tudíž 88 bodů na mobilu a 92 na počítači. Dobrých hodnot dosahuje i v dalších kategoriích, jako je přístupnost (93 bodů), dodržování doporučených postupů (96 bodů). Tato varianta exceluje i v SEO, kde dosáhla 100 bodů.

Pří vývoji za pomocí AI web překvapivě vykázal skvělou přístupnost, ve které získal podle PageSpeed Insights 100 bodů. Následně se jeho výkon snížil v případech mobilních zařízení i počítače, a to převážně kvůli příliš velké velikosti obrázků vygenerovaných pomocí AI.

Nejhorších výsledků ve všech testovaných metrikách dosáhl web vytvořený pomocí redakčního systému WordPress. Na mobilních zařízeních byla hodnota výkonu pouze 49 bodů, a přestože přístupnost a SEO nejsou zcela zanedbatelné, kombinace s nízkým hodnocením doporučených postupů signalizuje méně efektivní technické řešení. I na počítačích zůstává výkon nižší oproti ostatním přístupům. Výsledky ukazují na typické problémy CMS platform, které často generují rozsáhlý a méně optimalizovaný kód kvůli univerzálnosti a množství přídavných prvků.

Tabulka 4 - Hodnocení výkonu webové stránky na mobilních zařízeních (autorské zpracování)

Způsob vývoje/Ukazatel	Výkon	Přístupnost	Doporučené postupy	SEO
<b>Vývoj bez AI</b>	88	93	96	100
<b>Vývoj s AI</b>	69	100	96	91
<b>CMS vývoj</b>	49	94	75	83

Hodnocení v tabulce je formou bodů od 0 do 100.

Tabulka 5 - Hodnocení výkonu webové stránky na počítači (autorské zpracování)

Způsob vývoje/Ukazatel	Výkon	Přístupnost	Doporučené postupy	SEO
<b>Vývoj bez AI</b>	92	93	96	100
<b>Vývoj s AI</b>	91	100	96	91
<b>CMS vývoj</b>	69	94	78	83

Hodnocení v tabulce je formou bodů od 0 do 100.

## 2.4 Tvorba vizuálního obsahu

V rámci tvorby webových aplikací je vizuální stránka projektu klíčovým faktorem ovlivňujícím výslednou uživatelskou zkušenosť. Výběr a použití obrázků se tak stává nedílnou součástí celého vývojového procesu. Obrázky mohou být získávány různými způsoby, přičemž v této práci je analyzováno použití dvou přístupů: obrázky generované pomocí umělé inteligence a obrázky získané běžnou cestou prostřednictvím veřejných databází nebo vlastního focení.

Každý z těchto přístupů přináší specifické výhody i omezení, a to jak z hlediska kvality výsledného vizuálního materiálu, tak z pohledu technické náročnosti, časové investice a právních aspektů použití. Cílem této části je popsát proces tvorby a získávání obrázků, analyzovat jejich výsledné vlastnosti a provést srovnání z hlediska kvality, relevance k obsahu webu, velikosti souboru, časové náročnosti a právních podmínek použití.

Analýza bude vycházet z konkrétních příkladů použitých obrázků na webu a zaměří se na praktické aspekty volby vhodné metody tvorby vizuálního obsahu v kontextu vývoje webových projektů.

## 2.4.1 Obrázky získané bez použití AI

Pro potřeby tvorby webové prezentace byly obrázky získávány standardními způsoby bez využití generativní umělé inteligence. Tento proces zahrnoval tři hlavní přístupy: vlastní fotografování, využití volně dostupných databází a použití ikon ze specializovaných grafických zdrojů.

Část vizuálního obsahu byla vytvořena vlastní dokumentací provozu firmy. Byly pořízeny fotografie areálu, výrobního zařízení, expedice produktů a zaměstnanců. Tento přístup umožnil zachytit autentické prostředí firmy a zajistit relevantní a jedinečný vizuální materiál, který odpovídá skutečnému provozu a identitě společnosti.



Obrázek 29 - Fotografie expedice řeziva bez AI  
(autor)



Obrázek 30 - Fotografie štěpkování bez AI  
(autor)



Obrázek 31 - Fotografie expedice řeziva bez AI  
(autor)



Obrázek 32 – Fotografie štěpkování bez AI  
(autor)

Další skupinu obrázků tvoří grafické prvky stažené z volně dostupných galerií, například ilustrační fotografie přírodních materiálů. Příkladem je motiv dřevěných špalků ve tvaru srdce, použitý na Home page. Tento typ obsahu byl volen s ohledem na estetickou přitažlivost a tematickou relevanci k oboru dřevozpracující výroby.

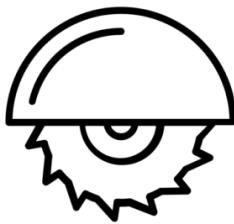


Obrázek 33 - Fotografie dřevěného srdce z fotobanky bez AI (autor)

Pro vizuální doplnění byly využity ikony získané ze specializovaných zdrojů, jako je platforma Flaticon, následně použité zejména v sekcích shrnujících klíčová fakta o firmě a sloužící k vizuálnímu zvýraznění informací a zlepšení přehlednosti prezentace, nebo pro tvorbu firemního loga. Firemní logo bylo autorem vytvořeno pomocí nástroje Canva, který má široké využití, mezi něž patří i tvorba firemního brandingu. Verze loga viditelná níže má inverzní barvu vůči verzi, která je viditelná na webu, kvůli lepší čitelnosti v textu.



Obrázek 34 - Ikona listnatého stromu bez AI (autor)



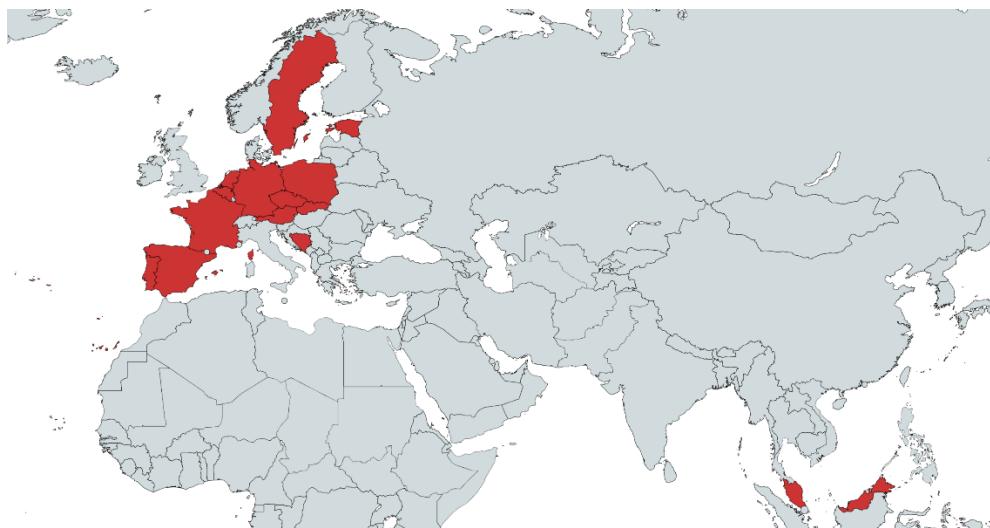
Obrázek 35 - Ikona pilového kotouče bez AI (autor)



Obrázek 36 - Logo firmy bez AI (autor)

Za účelem vyobrazení exportu výrobků byla navíc vytvořena vlastní mapa pomocí online nástroje MapChart. Tento nástroj umožňuje jednoduše generovat tematické mapy podle

zvolených oblastí a barevného označení. Výsledná mapa byla exportována ve vhodném formátu a sloužila k přehlednému zobrazení exportních destinací firmy.



Obrázek 37 - Mapa expedice bez AI (autor)

Při výběru všech obrázků bylo dbáno na kvalitu rozlišení, vhodnost formátu pro použití na webu a ověření licenčních podmínek, aby byl zajištěn legální způsob jejich užití. Všechny obrázky byly následně optimalizovány pro rychlejší načítání stránek, zejména úpravou rozlišení a velikosti souboru.

## 2.4.2 Obrázky generované pomocí AI

Další podkapitolou v rámci praktické části práce je možnost tvorby vizuálního obsahu prostřednictvím nástrojů využívajících umělou inteligenci. Tento přístup umožňuje generovat vlastní grafiku na základě textových zadání neboli promptů, bez nutnosti manuálního kreslení nebo fotografování.

Tato část práce se zaměřuje na popis procesu generování obrázků pomocí AI a jejich následné porovnání s odpovídajícími obrázky získanými tradičními metodami. Porovnávání bude rozděleno tematicky, a to zvlášť pro focené fotografie, fotobankové ilustrace ikony a generovanou mapu. U každé skupiny bude popsán proces generování, použité prompty a následně provedeno základní obsahové porovnání mezi vygenerovanými a získanými obrázky.

Obrázky v této podkapitole byly generovány za použití kombinace ChatGpt 4.5 a Open AI nástroje na generování obrázků a videa Sora.

### Fotografované obrázky

V této části byl simulován proces nahrazení reálných fotografií pořízených přímo v areálu firmy obrázky generovanými prostřednictvím nástrojů umělé inteligence. Cílem bylo vytvořit srovnatelné vizuály zachycující podobné motivy jako původní fotografie, a následně je vzájemně porovnat:

## Tým zaměstnanců

Použitý prompt: *Generate a photorealistic picture of about 20 employees in dirty work gear in a group in the working hall of a sawmill.*



Obrázek 38 - Fotografie tým s AI (autor) Obrázek 39 - Fotografie tým bez AI (autor)

Porovnání ukázalo, že AI dokázala vytvořit realisticky působící scénu se skupinou pracovníků, nicméně některé detailey v oblasti proporcí postav a výrazu obličejů působí uměle a očividně chybně.

## Sušárna řeziva

Použitý prompt: *Generate a photorealistic picture of an industrial wood drier suzar with a load of planks.*



Obrázek 40 - Fotografie sušárna s AI (autor)

Obrázek 41 - Fotografie sušárna bez AI (autor)

Výsledek odpovídal základnímu konceptu, avšak určité detailey, například konstrukční řešení budovy nebo značení, nebyly zcela realistické ve srovnání s reálnou sušárnou.

## Expedice řeziva

Použitý prompt: *Generate a photorealistic image of a european style truck carrying a load of planks in a ship container which is open from the back side.*



Obrázek 42 - Fotografie expedice řeziva s AI (autor)



Obrázek 43 - Fotografie expedice bez AI (autor)

Výstup působil velmi realisticky a není mu co vytknout.

## Štěpkování a manipulace s materiélem

Použitý prompt: *Generate a photorealistic picture of a manitou mlt filling up an industrial wood chipper forst.*



Obrázek 44 – Fotografie štěpkování s AI (autor)



Obrázek 45 - Fotografie štěpkování bez AI (autor)

Výsledný obrázek dokázal dobře vystihnout hlavní scénu, ale některé technické detaily na strojích nebo jeden stroj jako takový byly oproti původní fotografií zcela rozdílné.

## Mistr výroby

Použitý prompt: *Generate a corporate headshot of a production manager in his work clothes (jeans and tshirt) he is 45 bald and has no glasses.*



Obrázek 46 - Fotografie mistr s AI (autor)



Obrázek 47 - Fotografie mistr bez AI (autor)

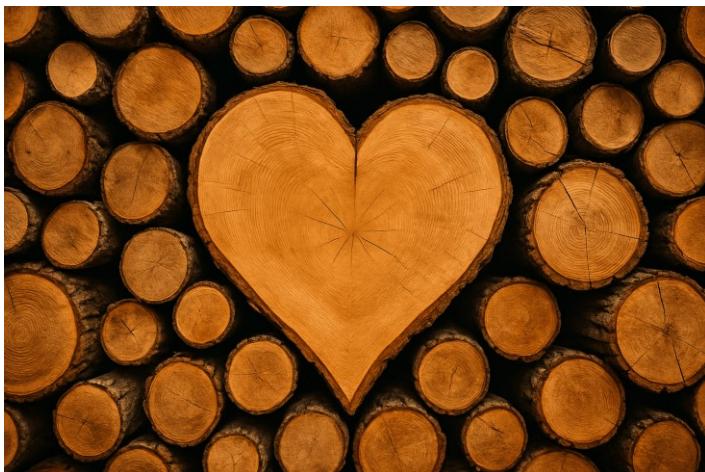
Výsledek generování odpovídal zamýšlenému tématu, přičemž celkové vyznění obrázku působilo dobře. Obrázek je odlišný v mnoha věcech, cílem ale nebylo zachytit vybraného zaměstnance, ale spíše typ obrázku.

## Fotografie z veřejně dostupných fotobank

Druhou oblastí, kde byly obrázky generovány pomocí umělé inteligence, byly ilustrace běžně získávané z veřejně dostupných fotobank. Cílem bylo vytvořit obrázek, který by tematicky a vzhledově navazoval na motiv dřevěného srdce, který je na webu používán v sekci hero.

## Dřevěné srdce ze špalků

Použitý prompt: *Generate one big log in shape of a heart which is surrounded by other smaller logs in the background.*



Obrázek 48 - Fotografie dřevěného srdce s AI (autor)



Obrázek 49 - Fotografie dřevěného srdce z fotobanky bez AI (autor)

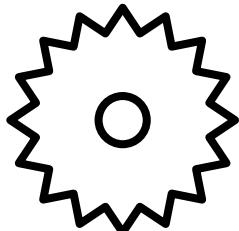
Výstup dokázal zachytit požadovanou atmosféru, ale některé proporce okolních kmenů a struktura středového kmene naznačovaly umělou repetitivní kompozici typickou pro generované obrázky.

## Ikony a grafické prvky

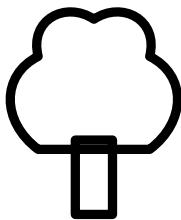
Pro zlepšení orientace na webu bylo také třeba vygenerovat ikony, jež byly v prvním případě získány z veřejné databáze free ikon jménem Flaticon. Jedna z těchto ikon byla rovněž použita k vytvoření původního loga firmy, tudíž bylo do této kategorie zařazeno i vygenerování nového loga firmy.

## Ikony pilového kotouče a listnatého stromu

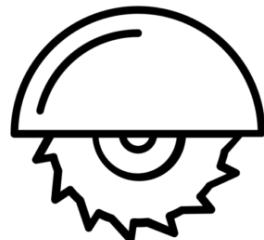
Použité prompty: *Generate svg icons, which represent in order a saw blade, an oak tree, a spruce tree and a ruler.*



Obrázek 50 - Ikona pilového kotouče s AI (autor)



Obrázek 51 - Ikona listnatého stromu s AI (autor)



Obrázek 52 - Ikona pilového kotouče bez AI (autor)



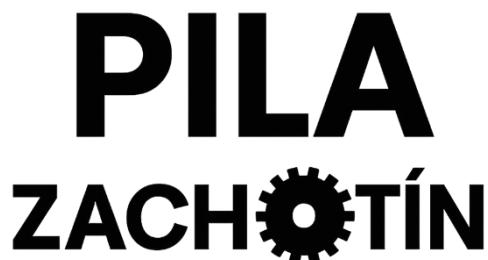
Obrázek 53 - Ikona listnatého stromu bez AI (autor)

Výsledné ikony vizuálně navazovaly na tematiku webu, nicméně ve srovnání s profesionálně připravenými ikonami z databáze jako Flaticon působily výrazně jednodušeji a méně detailně, což může ve výsledku ovlivnit důvěryhodnost webu.

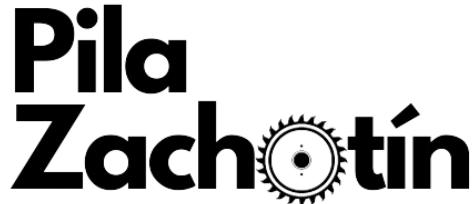
## Logo společnosti

Použitý prompt: *Generate a logo for a company called Pila Zachotín. It is a sawmill. The logo should be white with low detail. It should look like: row1 – Pila, row2 – Zachotín, but replace the o with a nice professional looking saw icon and center it so it looks like it fits in the text (the saw icon can go below the text line for the purpose of centering it perfectly with the word Zach(saw)tín). The text in the logo should be centered left.*

Pro účely snažšího čtení textu jsou obě loga v inverzní barvě k barvě původní – bílá.



Obrázek 54 - Logo firmy s AI (autor)



Obrázek 55 - Logo firmy bez AI (autor)

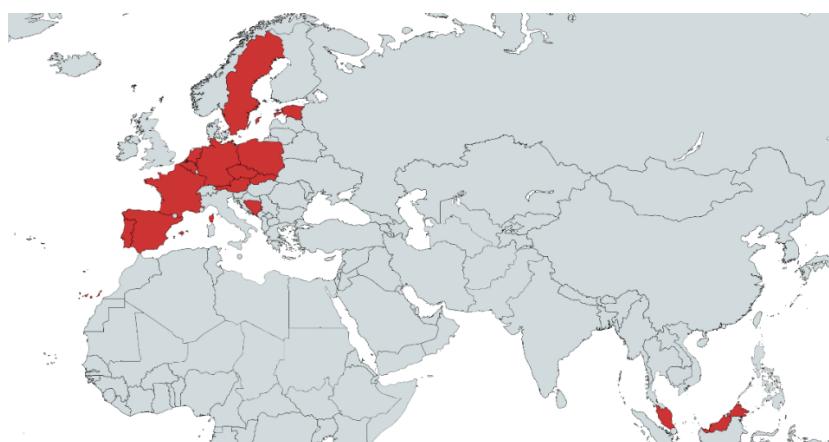
Výsledné logo reflektovalo požadavek na zařazení pilového kotouče a dřevozpracujícího tématu, ale vizuální provedení působilo méně konzistentně a neodpovídalo plně původnímu vzhledu.

## Mapové znázornění exportních destinací

Použitý prompt: *Generate a map of the world which is visibly grey on a white background. Fill in these countries with red color: (czechia, poland, slovakia, germany, bosnia and hercegovina, spain, portugal, france, netherlands, belgium, austria, sweden, estonia and malaysia)*



Obrázek 56 - Mapa expedice s AI (autor)



Obrázek 57 - Mapa expedice bez AI (autor)

Výsledná mapa odpovídala základnímu požadavku na vyznačení příslušných států, až na Francii, avšak některé detaily v rozlišení hranic a barevném ladění se odlišovaly od map generovaných pomocí nástroje MapChart. Mapa vytvořená umělou inteligencí byla zdeformovaná, na rozdíl od mapy generované pomocí nástroje MapChart, která byla geograficky přesná.

### 2.4.3 Souhrn

#### Relevance a estetika

Pro hodnocení relevance obrázků byla analyzována pouze skupina obrázků generovaných pomocí umělé inteligence. Důvodem bylo, že tradičně získané obrázky (fotografie a grafické

prvky) byly pořizovány přímo s ohledem na maximální tematickou vhodnost vzhledem k obsahu webu, a tedy jejich relevance nebyla předmětem dalšího hodnocení.

U obrázků generovaných umělou inteligencí byla relevance hodnocena pomocí jednoduchého systému, kde obrázky byly zařazeny do kategorií dostačující nebo nedostačující. Kritériem bylo, zda vygenerovaný obsah odpovídá zamýšlenému významu a kontextu použití na webu. Obrázky označené jako „dostačující“ byly považovány za dostatečně vystihující dané téma, zatímco „nedostačující“ obrázky vykazovaly odchylky, které by mohly narušit správné pochopení obsahu.

Tabulka 6 - Relevance vizuálního obsahu (autorské zpracování)

Obrázek	Relevance
<b>Tým zaměstnanců</b>	Nedostačující
<b>Expedice dřeva</b>	Dostačující
<b>Sušárna řeziva</b>	Dostačující
<b>Štěpkování a manipulace s materiélem</b>	Dostačující
<b>Mistr výroby</b>	Dostačující
<b>Dřevěné srdce ze špalků</b>	Dostačující
<b>Ikona pilového kotouče</b>	Dostačující
<b>Ikona listnatého stromu</b>	Dostačující
<b>Logo firmy</b>	Dostačující
<b>Mapové znázornění exportních destinací</b>	Nedostačující

Tým zaměstnanců byl označen jako nedostačující, jelikož kvalita vygenerovaného obrázku nevykresluje obličeje zaměstnanců v dostatečné kvalitě a budí u nich nelidský dojem. Mapové znázornění exportu bylo označeno jako nedostačující, protože rozložení mapy je příliš abstraktní a chybí v něm jedna z požadovaných destinací exportu.

Hodnocení estetické kvality obrázků bylo provedeno na základě promptu vytvořeného pomocí nástroje ChatGPT. Tento prompt specifikoval základní kritéria, podle kterých byly obrázky porovnávány z hlediska vizuální přitažlivosti, kompozice, barevného ladění a celkového dojmu.

Použitý prompt pro hodnocení estetiky: *Posud' estetickou kvalitu dvojic obrázků (klasický vs. AI) podle těchto kritérií: celková vizuální přitažlivost, kompozice, práce se světlem a barevným kontrastem. U každé dvojice urči, který obrázek lépe splňuje tato kritéria.*

Výsledky hodnocení byly shrnuty v přehledné tabulce, kde bylo pro každý obrázek určeno, zda z hlediska estetiky vítězí klasický obrázek, nebo obrázek vytvořený pomocí umělé inteligence.

Obrázek	Estetika
Tým (skupinová fotografie zaměstnanců)	klas.
Mistr výroby (portrét)	AI
Sušárna řeziva	klas.
Expedice řeziva	AI
Štěpkování (manipulace s dřevem)	klas.
Dřevěné srdce (wooden heart)	AI
Ikona pilového kotouče	klas.
Ikona stromu	klas.
Logo společnosti	klas.
Mapa exportu (geografické znázornění)	klas.

Poznámka: "klas." značí vítězství klasického (získaného) obrázku, "AI" značí vítězství vygenerovaného AI obrázku v kategorii estetiky.

Obrázek 58 - Výstup z ChatGPT 4.5 (autor)

### Kvalita výstupu

Technická kvalita obrázků byla v rámci této práce posuzována především na základě rozlišení v pixelech a velikosti souboru, což jsou dva základní technické parametry ovlivňující ostrost, detailnost a použitelnost obrázků na webu. Hodnocení probíhalo porovnáním dvojic klasických a umělou inteligencí generovaných vizuálů.

U tradičně získaných obrázků byla technická kvalita na vysoké úrovni, a to zejména díky jejich pořízení pomocí profesionální techniky a následné optimalizaci pro webové prostředí. Fotografie byly pořízeny ve vysokém rozlišení, a následně zmenšeny tak, aby si při udržení

kvality zachovaly optimální datovou velikost pro rychlé načítání bez nadbytečné komprese. Výsledkem byly soubory s vyváženým poměrem mezi ostrostí, velikostí a funkčností.

Naopak u obrázků vytvořených pomocí nástrojů umělé inteligence se objevily specifické odlišnosti. Ačkoli většina těchto obrázků byla vygenerována ve velmi vysokém rozlišení, jejich souborová velikost byla často neúměrně velká, v některých případech několikanásobně vyšší než u optimalizovaných klasických obrázků. To může negativně ovlivnit dobu načítání stránek i celkovou efektivitu webu. Ve výjimečných případech se u AI obrázků vyskytly i drobné vizuální nedostatky, například jemné artefakty či neostré detaily v okrajových částech obrazu.

Technické srovnání jednotlivých obrázků bylo shrnuto v přehledné tabulce, kde bylo u každé dvojice určeno, který z obrázků dosahuje vyšší úrovně technické kvality na základě výše uvedených parametrů.

Tabulka 7 - Kvalita rozlišení vizuálního obsahu (autorské zpracování)

Obrázek – rozlišení	AI	Klasické
<b>Tým zaměstnanců</b>	1Mpx	0.3Mpx
<b>Expedice dřeva</b>	1.6Mpx	0.3Mpx
<b>Sušárna řeziva</b>	1Mpx	0.3Mpx
<b>Štěpkování a manipulace s materiélem</b>	1.6Mpx	0.3Mpx
<b>Mistr výroby</b>	1.6Mpx	0.3Mpx
<b>Dřevěné srdce ze špalků</b>	1.6Mpx	1.2Mpx
<b>Ikona pilového kotouče</b>	0.02Mpx	0.1Mpx
<b>Ikona listnatého stromu</b>	0.02Mpx	0.1Mpx
<b>Logo firmy</b>	0.1Mpx	0.25Mpx
<b>Mapové znázornění exportních destinací</b>	1.4Mpx	4.8Mpx

Tabulka 8 - Kvalita velikosti vizuálního obsahu (autorské zpracování)

Obrázek – velikost	AI	Klasické
<b>Tým zaměstnanců</b>	1.52MB	0.13MB
<b>Expedice dřeva</b>	2.41MB	0.13MB
<b>Sušárna řeziva</b>	1.78MB	0.08MB
<b>Štěpkování a manipulace s materiélem</b>	2.79MB	0.13MB
<b>Mistr výroby</b>	2.33MB	0.11MB
<b>Dřevěné srdce ze špalků</b>	1.24MB	0.7MB
<b>Ikona pilového kotouče</b>	0.6kB	10kB
<b>Ikona listnatého stromu</b>	0.6kB	10kB
<b>Logo firmy</b>	1.1kB	38kB
<b>Mapové znázornění exportních destinací</b>	0.26MB	0.96MB

### Právní a etický aspekt

Při tvorbě vizuálního obsahu pomocí nástrojů umělé inteligence v rámci této práce se objevily konkrétní právní i etické problémy, které je nutné zmínit.

Z právního hlediska představovaly určitou komplikaci vygenerované obrázky obsahující rozpoznatelné značky výrobců, loga nebo charakteristické prvky konkrétních produktů. Vzhledem k tomu, že tyto značky a designy mohou být chráněny autorskými právy nebo ochrannými známkami, vznikla nejistota ohledně legálního použití těchto obrázků na veřejných webových stránkách. Při použití obrázků s těmito znaky by bylo potřeba zvážit, zda jejich publikace nepředstavuje porušení práv duševního vlastnictví.

Z etického hlediska se při generování portrétních fotografií projevily typické nedostatky související s deformacemi obličejobých rysů. Některé vygenerované postavy vykazovaly nerealistické nebo zkreslené proporce, což by v případě použití na webu mohlo vést k nevhodné prezentaci firmy nebo k vyvolání nedůvěry u návštěvníků. Dále nebyly v některých případech jasně označeny obrázky jako „AI generated“, což by mohlo být z etického hlediska problematické, neboť by návštěvník nemusel být informován, že se nejedná o skutečný fotografický záznam reálné situace.

## 2.5 Multikriteriální porovnání přístupů

### 2.5.1 Zvolení cílů

Cílem multikriteriálního porovnání je učinit rozhodnutí na základě zjištěných informací týkající se toho, v jakých částech tvorby webové stránky je lepší zvolit řešení s umělou inteligencí a v jakých aplikovat tradiční přístupy.

### 2.5.2 Výpočet vah kritérií

Kritéria se porovnávají párově, a to tak, že každé kritérium je porovnáno s každým jiným kritériem podle své vzájemné důležitosti. Hodnoty důležitosti jsou určeny škálou 1-9 podle pravidel Saatyho metody:

Tabulka 9 - Hodnocení kritérií podle Saatyho metody (autorské zpracování)

#### HODNOTA KRITÉRIA

1	Kritéria jsou stejně důležitá
3	Kritérium v řádku je mírně důležitější než v sloupci
5	Kritérium v řádku je výrazně důležitější.
7	Kritérium v řádku je velmi silně preferováno
9	Kritérium v řádku je absolutně preferováno

### Výpočet vah kritérií pro vývoj kódu.

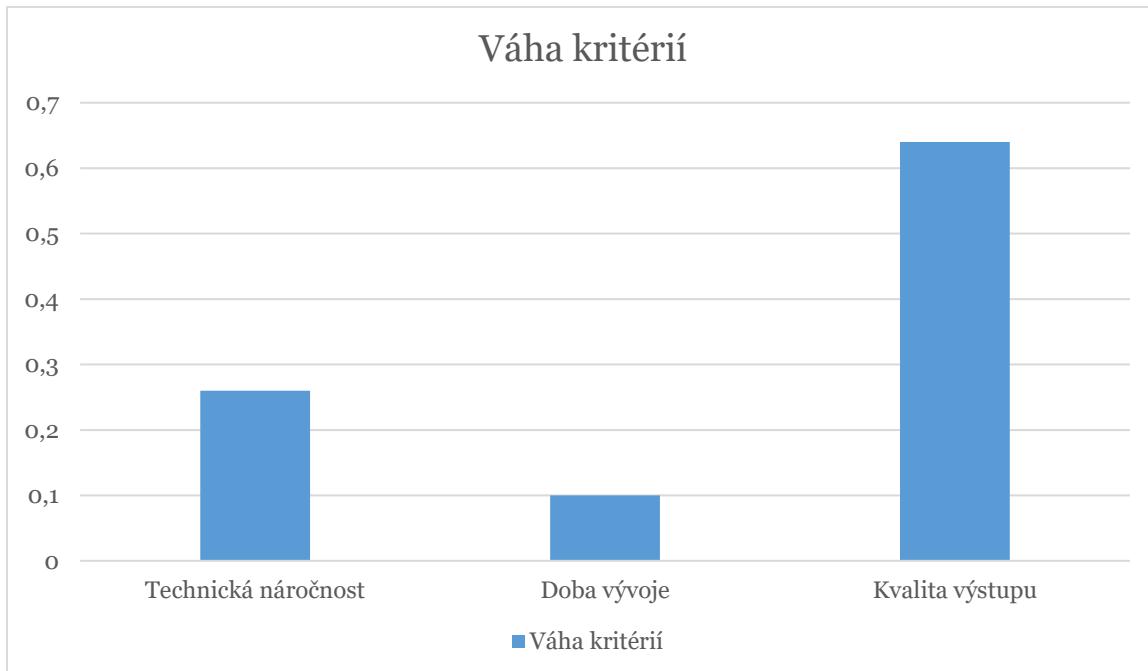
Tabulka 10 - Párové srovnání a výpočet vah kritérií pro vývoj kódu (autorské zpracování)

Kritéria	Technická náročnost	Doba vývoje	Kvalita výstupu	Geom. průměr	Váha
Technická náročnost	1	3	0.33	1.00	0.26
Doba vývoje	0.33	1	0.2	0.41	0.10
Kvalita výstupu	3	5	1	2.47	0.64

Tato tabulka představuje výsledné váhy tří klíčových kritérií použitých pro hodnocení metod vývoje webového kódu: Technická náročnost, Doba vývoje a Kvalita výstupu. Nejvyšší váhu

získala Kvalita výstupu (0,64), což reflektuje její zásadní význam pro celkové posouzení úrovně jednotlivých řešení. Následuje Technická náročnost (0,26), která vyjadřuje míru odbornosti potřebné pro úspěšnou implementaci, a nakonec Doba vývoje (0,1), jakožto faktor související s efektivitou práce.

Celkový poměr konzistence této maticy činí přibližně 3,4 %<sup>6</sup>, což značí, že hodnocení je v rámci metody párového porovnání relativně logické a bez významných vnitřních nesrovnalostí. Grafické znázornění vah použitých kritérií lze vidět na přiloženém grafu.



Obrázek 59 - Grafické znázornění vah stanovených kritérií pro vývoj kódu (autorské zpracování)

### **Výpočet vah kritérií pro získávání obrázků.**

Tabulka 11 - Párové srovnání a výpočet vah kritérií pro tvorbu vizuálního obsahu (autorské zpracování)

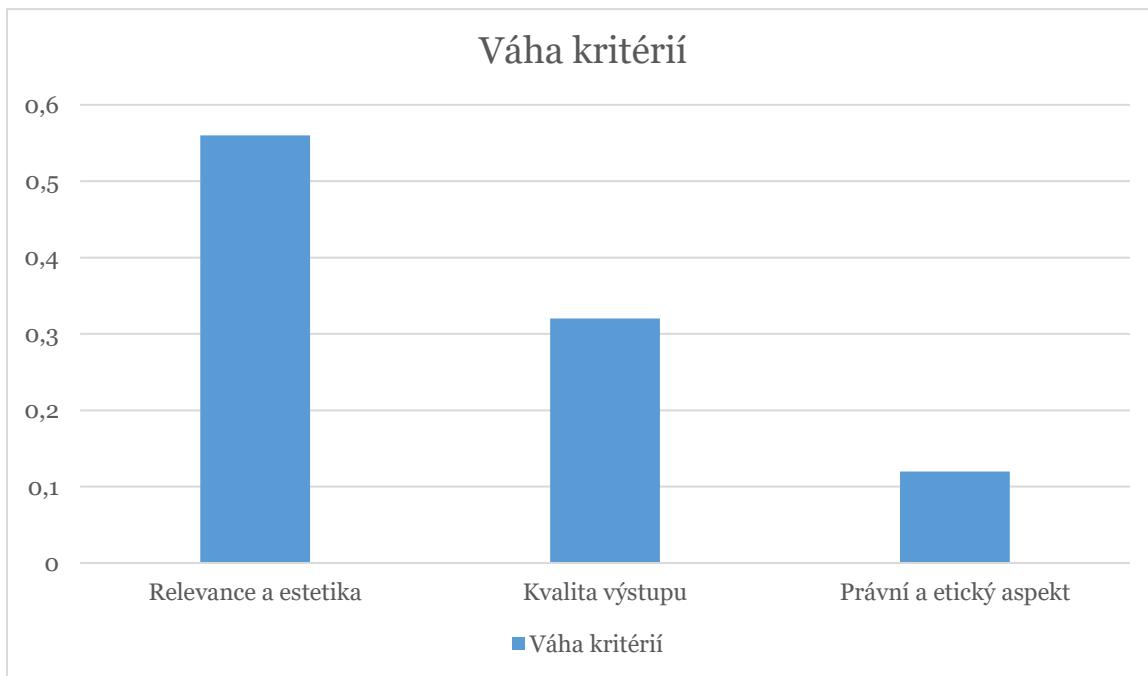
Kritéria	Relevance a estetika	Kvalita výstupu	Právní a etický aspekt	Geom. průměr	Váha
<b>Relevance a estetika</b>	1	2	4	2	0.56
<b>Kvalita výstupu</b>	0.5	1	3	1.14	0.32

<sup>6</sup> Spočítáno pomocí <https://bpmsg.com/ahp/ahp-calc.php>

Kritéria	Relevance a estetika	Kvalita výstupu	Právní a etický aspekt	Geom. průměr	Váha
Právní a etický aspekt	0.25	0.33	1	0.44	0.12

Tato tabulka představuje konečné váhy tří klíčových kritérií pro hodnocení kvality obrázků: Relevance a estetika, Kvalita výstupu a Právní a etický aspekt. Největší váhu získalo kritérium Relevance a estetika s hodnotou 0,56, což odráží jeho dominantní roli v celkovém vnímání kvality vizuálního obsahu. Následuje Kvalita výstupu s vahou 0,32, která zohledňuje technické vlastnosti jako rozlišení a velikost souboru. Nejnižší váhu, 0,12, získalo kritérium Právní a etický aspekt, což ukazuje na jeho relativně menší vliv při rozhodování, přestože zůstává důležitým faktorem z hlediska odpovědnosti a souladu s legislativou.

Poměr konzistence této matice je 1,9 %, což potvrzuje vysokou míru logické konzistence mezi jednotlivými párovými porovnáními. Hodnoty byly stanoveny tak, aby co nejvěrněji odrážely praktické zkušenosti s jednotlivými kritérii během experimentální části práce.



Obrázek 60 - Grafické znázornění vah stanovených kritérií pro tvorbu vizuálního obsahu (autorské zpracování)

## **2.5.3 Porovnání přístupů**

### **Porovnání z hlediska vývoje kódu**

#### **Technická náročnost**

Ruční vývoj bez využití nástrojů umělé inteligence byl jednoznačně nejnáročnější, a to jak z hlediska odborných znalostí, tak i množství dovedností potřebných k efektivnímu zvládnutí práce. Autor musel ovládat strukturu HTML, stylování pomocí CSS i interaktivní prvky v JavaScriptu.

Naopak vývoj pomocí AI snižoval potřebu těchto znalostí, ale ne výrazně. Velká část kódu byla sice automaticky vygenerována na základě zadaných promptů, přesto však bylo nezbytné kontrolovat výstupy, rozumět jim a často provádět opravy.

Nejnižší technickou náročnost vykazoval CMS systém WordPress, který umožňuje tvorbu webu bez nutnosti psaní kódu, a to pouze za pomocí vizuálního editoru, předpřipravených šablon a upravování struktury jejich prvků.

Hodnocení tradičního vývoje: 3

Hodnocení vývoje s AI: 5

Hodnocení CMS řešení: 9

#### **Doba vývoje**

Z časového hlediska byl tradiční způsob tvorby webu nejpomalejší, s nutností vše ručně implementovat a testovat, což zabralo celkem 19 hodin. Vývoj s pomocí AI umožnil výrazné zrychlení, a jelikož byl kód generován, zkrátil se pracovní čas na polovinu, tedy přibližně na 9 hodin. Nejrychlejším přístupem bylo využití CMS řešení, které díky hotovým blokům a intuitivnímu rozhraní umožnilo vytvořit web za pouhé 4 hodiny.

Hodnocení tradičního vývoje: 3

Hodnocení vývoje s AI: 6

Hodnocení CMS řešení: 8

#### **Kvalita výstupu**

Při porovnání kvality výsledného webu se ukázalo, že ručně vytvořený kód měl nejlepší skóre jak ve validaci HTML (W3C), tak v rychlosti načítání a optimalizaci pro SEO. AI výstup měl rovněž kvalitní strukturu a přehlednost, ale místy se objevily chyby související s se strukturou HTML nebo nedostatečnou optimalizací rychlosti načítání. CMS řešení vykazovalo největší počet chyb v HTML validátorech, méně optimalizovaný kód a nižší výkon kvůli příliš komplikované stavbě systému šablon vzhledem k jednoduchosti webového řešení.

Hodnocení tradičního vývoje: 8

Hodnocení vývoje s AI: 7

Hodnocení CMS řešení: 4

Tabulka 12 - Hodnocení přístupů k vývoji z hlediska zvolených kritérií před započtením vah (autorské zpracování)

	Technická náročnost	Doba vývoje	Kvalita výstupu
Vývoj bez AI	3	3	8
Vývoj s AI	5	6	7
CMS vývoj	9	8	4

V tabulce výše můžeme vidět ohodnocení každého kritéria pro všechny tři přístupy k vývoji. Každé hodnocení v této tabulce bylo vynásobeno adekvátní váhou kritéria, která vznikla z předchozího párového porovnání a celkové výsledky byly následně sečteny do výsledného kombinovaného skóre pro každou metodu. Tento proces je reprezentován v tabulce níže.

Tabulka 13 - Hodnocení přístupů k vývoji z hlediska zvolených kritérií po započtení vah (autorské zpracování)

	Technická náročnost	Doba vývoje	Kvalita výstupu	Celkové skóre
Váha	0.26	0.1	0.64	1
Vývoj bez AI	0.78	0.3	5.12	6.2
Vývoj s AI	1.3	0.6	4.48	6.38
CMS vývoj	2.34	0.8	2.56	5.7

Celkové skóre pro vývoj bez umělé inteligence vyšlo 6,2, pro vývoj s AI 6,38 a pro vývoj za užití CMS 5,7. V důsledku těchto výsledných hodnot můžeme usoudit, že vývoj kódu s použitím AI je nejvhodnější ze tří variant.

### Porovnání z hlediska tvorby vizuálního obsahu

#### Relevance a estetika

Tradičně získané obrázky byly pořizovány s ohledem na konkrétní potřeby webu, a jejich tematická vhodnost tedy odpovídala zamýšlenému účelu. V případě AI generovaných vizuálů bylo třeba ověřit, zda vytvořený obsah skutečně reflektuje požadované sdělení. Relevance byla proto posuzována pouze u obrázků vytvořených pomocí AI. Většina vizuálů splnila očekávání, avšak v některých případech byla relevance vyhodnocena jako

nedostatečná. Z hlediska estetiky výsledky ukázaly, že AI často produkuje esteticky atraktivní vizuály, obzvláště u ilustračních nebo stylizovaných záběrů. I přesto ale bylo v převaze množství klasicky získaných obrázků, které byly vyhodnoceny za esteticky přívětivější.

Hodnocení tradičně získaných obrázků: 8

Hodnocení AI generovaných obrázků: 6

### **Kvalita výstupu**

Technická kvalita byla hodnocena především na základě rozlišení (v pixelech) a velikosti výsledného souboru. U tradičních fotografií bylo dosaženo dobrého poměru mezi ostrostí a optimalizací pro web, jelikož byly obrázky zpracovány tak, aby měly co nejnižší datovou velikost při zachování vizuální kvality.

AI obrázky byly často vygenerovány ve vysokém rozlišení, avšak jejich velikost bývala výrazně vyšší, což negativně ovlivnilo rychlosť načítání stránek.

Podobně dopadly i prvky ikon, kde klasická řešení nabídla ostřejší a lépe optimalizovaný výstup.

Hodnocení tradičně získaných obrázků: 7

Hodnocení AI generovaných obrázků: 6

### **Právní a etický aspekt**

Tradičně získané obrázky byly pořizovány s vědomím autorských a etických pravidel, a jejich použití na webu nepředstavovalo žádné riziko. U AI generovaného obsahu se objevily určité problémy: některé vizuály zobrazovaly znaky nebo prvky, které by mohly být chráněny autorským právem, a v portrétech docházelo k nepřirozenému zkreslení obličejů.

Další etický problém vznikající v souvislosti s označením obrázků je absence jasné identifikace, že se jedná o výstup vytvořený umělou inteligencí. Tento fakt by měl být jasně oznámen v souvislosti s používáním generovaných obrázků, a to by mohlo firmu poškodit v oblasti brand image.

Hodnocení tradičně získaných obrázků: 9

Hodnocení AI generovaných obrázků: 4

Tabulka 14 - Hodnocení přístupů k tvorbě vizuálního obsahu z hlediska zvolených kritérií před započtením vah (autorské zpracování)

	<b>Relevance a estetika</b>	<b>Kvalita výstupu</b>	<b>Právní a etický aspekt</b>
<b>Klasický způsob</b>	8	7	9

	<b>Relevance a estetika</b>	<b>Kvalita výstupu</b>	<b>Právní a etický aspekt</b>
<b>Generování pomocí AI</b>	6	6	4

Výše můžeme vidět vyhodnocení kritérií pro oba přístupy k získávání obrázků. Všechna tato hodnocení byla vynásobena váhou zvolených kritérií, která vznikla při párovém porovnávání. Tímto způsobem bylo utvořeno skóre, uvedené v tabulce níže.

Tabulka 15 - Hodnocení přístupů k tvorbě vizuálního obsahu z hlediska zvolených kritérií po započtení vah (autorské zpracování)

	<b>Relevance a estetika</b>	<b>Kvalita výstupu</b>	<b>Právní a etický aspekt</b>	<b>Celkové skóre</b>
<b>Váha</b>	0.56	0.32	0.12	1
<b>Klasický způsob</b>	4.48	2.24	1.08	7.8
<b>Generování pomocí AI</b>	3.36	1.92	0.48	5.76

Výsledné skóre pro získávání obrázků prostřednictvím tradičních metod je 7,8 zatímco skóre pro obrázky generované za asistence umělé inteligence je 5,76. Z těchto výsledků můžeme usoudit, že tradiční přístup je výrazně výhodnější oproti generování za pomocí umělé inteligence. Tento závěr může mít mnoho důvodů, jako například opomenutí hodnocení kritéria flexibility, tedy možnosti generování bez závislosti na lokaci a personálu, nýbrž stále je očividné, že použití AI v mnoha směrech zaostává.

## 2.5.4 Výsledky

Na základě výsledků multikriteriálního porovnání lze říct, že využití AI v určitých aspektech vývoje webových aplikací přináší výrazné výhody, zatímco v jiných oblastech má stále značné limity.

V oblasti vývoje kódu se jako mírně výhodnější varianta oproti ostatním ukázalo použití generativní umělé inteligence. Tento přístup výrazně zkrátil dobu vývoje, výsledný kód byl často strukturován podle aktuálních standardů a reflektoval moderní přístupy, například škálovatelnost nebo modularitu. Přesto bylo zapotřebí určité úrovně odborných znalostí, a to zejména při ladění, kontrole výstupů a jejich úpravě. AI nástroje zatím nedokážou přesně porozumět kontextu zadání a ve složitějších případech generují neúplné nebo nadbytečné výstupy, které je nutné ručně upravovat.

Ruční vývoj, ačkoliv nejvíce časově i technicky náročný, vykázal nejlepší výsledky z hlediska kvality výstupu. Autor měl plnou kontrolu nad každým prvkem, což umožnilo vytvořit optimalizovaný, validní a výkonný web, který obstál i při podrobné technické analýze. Tato

metoda je však vhodná spíše pro zkušené vývojáře, kteří disponují potřebným know-how a časem.

Nejnižší technickou náročnost a zároveň nejkratší dobu realizace přineslo CMS řešení. Pomocí vizuálního editoru, předdefinovaných bloků a šablon bylo možné web sestavit bez nutnosti znalosti kódování. Výsledkem byl funkční web s minimálními nároky na odbornost, avšak kvůli automatizovanému způsobu stavění kódu se objevily problémy s kvalitou výstupu, především v podobě horšího skóre v HTML validaci, nižší výkonnosti a omezené možnosti optimalizace.

V oblasti práce s obrázky se tradičně pořízené a zpracované vizuály ukázaly jako výhodnější. Díky pečlivému výběru, tematické relevanci a kvalitní postprodukci měly nejen vyšší estetickou úroveň, ale i technickou optimalizaci pro webové prostředí. Naproti tomu AI generované obrázky přinášely vyšší rozlišení a vizuálně poutavý styl, nicméně často trpěly přehnanou velikostí souborů, občasné nerealističnosti a technickými nebo etickými problémy, například zkreslenými portréty či nejasným původem zobrazených prvků.

Závěrem lze říct, že AI představuje výrazný přínos pro urychlení a zjednodušení některých částí webového vývoje, zejména při generování textového kódu nebo návrhu vizuální struktury. Je však třeba ji kombinovat s odbornou kontrolou a korekcemi s cílem dosažení plně profesionálního výsledku. V oblasti vizuálního obsahu zatím AI nedokáže zcela nahradit tradiční přístupy, a to především kvůli otázkám kvality, relevance a právní jistoty. V budoucnu lze očekávat, že s dalším vývojem těchto nástrojů bude jejich praktické využití stále širší a efektivnější, nicméně i nadále zůstane klíčová role lidského zásahu při jejich použití.

# Závěr

Bakalářská práce se věnovala porovnání tří různých přístupů k vývoji webových stránek: ručnímu kódování, generování pomocí umělé inteligence a využití CMS systému WordPress. Teoretická část se zaměřila na historickou stránku vývoje, průběh vývoje a problémy, které při něm vznikají. Otevřela také otázku využití umělé inteligence pro vývoj webů. Praktická část obsahovala sérii experimentů zaměřených na vývoj webového kódu a tvorbu vizuálního obsahu webu, což přineslo detailní pohled na omezení a benefity AI ve vývoji webových aplikací.

Jedním z nejvýraznějších přínosů umělé inteligence je urychlení procesu vývoje kódu. Pomocí promptování bylo možné rychle vygenerovat funkční části HTML, CSS i JavaScriptu, což vedlo k výraznému zkrácení času potřebného k vytvoření webu. Tento přístup však vyžadoval určitou úroveň znalostí, protože generovaný kód v některých případech obsahoval chyby, neodpovídal přesně zadání nebo vykazoval nižší míru optimalizace. AI tak usnadnila počáteční fáze vývoje, ale zároveň vytvořila potřebu následné ruční kontroly a úprav. Přesto se ukázalo, že AI dokáže urychlit proces vývoje za souběžného udržení dostačující kvality kódu.

V oblasti práce s obrázky byly tradičně získané vizuály hodnoceny jako kvalitnější a tematicky přesnější. Obrázky generované pomocí AI byly naopak často esteticky zajímavé, ale v některých případech nedostatečně relevantní nebo technicky nevhodné pro použití na webu. Rovněž se objevily etické a právní otázky týkající se autentičnosti portrétů a použití chráněných značek.

Použití Saatyho metody umožnilo kvantifikovat výhody i nedostatky jednotlivých přístupů na základě vážených kritérií. Analýza ukázala, že v kontextu této práce a jejich limit je nejvýhodnějším přístupem z hlediska vývoje kódu zvoleného webu použití AI, jelikož efektivně vyvažuje dostačující úroveň kvality výstupu se střední technickou náročností a dobou vývoje. Na druhou stranu při tvorbě vizuálního obsahu zůstává stále tradiční metoda nejsbolehlivější volbou. Obrázky vygenerované umělou inteligencí pořád nedosahují dostačné kvality v porovnání s reálnými fotografiemi.

Závěrem lze říct, že umělá inteligence představuje perspektivní nástroj pro moderní webový vývoj, zejména tam, kde je klíčová rychlosť a efektivita. Zatím však nedokáže plně nahradit lidský zásah, zejména v oblastech, kde je potřeba kreativní citlivost, technická přesnost nebo právní odpovědnost. Budoucnost webového vývoje tak pravděpodobně spočívá v efektivní spolupráci člověka s AI, kde člověk definuje záměr a AI pomáhá s jeho realizací.

# Diskuse

Tato kapitola si klade za cíl zasadit zjištění a výsledky práce do širšího kontextu vývoje webových technologií. V první části jsou na základě získaných poznatků formulována doporučení pro efektivní využití nástrojů umělé inteligence ve webovém vývoji. V další části následuje diskuse týkající se metodologických a obsahových omezení, která ovlivňují přesnost a přenositelnost závěrů. Pozornost je v této části věnována jak omezenému rozsahu výzkumu, tak specifikům zvoleného zadání. V závěrečné části je pak nastíněn potenciál dalšího výzkumu a možné směry rozšíření této práce.

## Doporučení

V rámci diskuse lze z výsledků získaných touto prací a při vědomí jejich limit vyvodit několik doporučení pro vývojáře uvažující o využití AI pro vývoj webů:

- Používat AI pro tvorbu obecných požadavků, nikoliv konkrétních řešení:**  
AI bylo velmi nápomocné při ulehčení tvorby jednoduchých a repetitivních částí webu, ale v případě zaměření se na konkrétní požadavky a specifikace nebylo schopno dodat relevantní, konkrétně požadovaný výsledek.
- Nepoužívat AI jako náhradu za vývojáře:** Umělá inteligence není ve svém momentálním stavu stoprocentně přesná a vytváří výsledky, které nedopovídají svým zadáním, a proto je vždy třeba kontroly ze strany vývojáře.
- Kombinace tradičního vývoje s použitím AI:** Nejlepších výsledků lze dosáhnout při kombinaci tradičního řešení, které obdrželo nejlepší hodnocení z hlediska kvality, a řešení s AI, které výrazně urychlilo a zefektivnilo proces vývoje.

Momentální forma AI prokazuje velký potenciál ve změně tradičních procesů vývoje webových aplikací a webů, její aplikace však musí být důkladně zanalyzovány a připojeny k procesům, které fungují na tradiční bázi.

## Limity práce

Tato práce má několik omezení, která do určité míry ovlivňují rozsah a hloubku zkoumání problematiky využití umělé inteligence při vývoji webových aplikací. Mezi tato omezení patří zejména následující oblasti:

### Omezený rozsah použitých přístupů k tvorbě webu

Práce se zaměřuje na tradiční ruční kódování, vývoj s využitím AI nástrojů a tvorbu prostřednictvím CMS systému. Nezahrnuje však další relevantní přístupy, jako jsou pokročilé NoCode platformy typu Bubble nebo Webflow, které dnes představují samostatnou kategorii vývoje. Tyto nástroje by mohly nabídnout odlišné výsledky v oblasti časové efektivity, technické náročnosti i kvality výstupu. Jejich nezařazení tak do určité míry omezuje komplexnost celkového srovnání.

### Složitost vzorového webu

Závěry uvedené v této práci vycházejí z analýzy vývoje jednoho konkrétního webového projektu s předem definovanou úrovní složitosti. Výsledky proto nelze bez dalšího zobecňování vztahovat na celý obor vývoje webových aplikací ani na komplexní softwarová řešení, jako jsou rozsáhlé podnikové systémy, webové aplikace s pokročilou logikou nebo e-

commerce platformy. Je proto důležité vnímat výstupy této práce v kontextu zvolené úrovně projektu viz vzorový web.

### **Volba kritérií pro porovnání kvality výstupu**

Kritéria použitá pro hodnocení kvality výstupu poskytují užitečný rámec pro základní srovnání, avšak nezahrnují některé další aspekty, které mohou výrazně ovlivnit celkové hodnocení kvality. V případě kódu nebylo například zohledněno zabezpečení nebo přehlednost kódu pro další kooperaci. U obrázků pak nebyla například posuzována konzistence vizuálního stylu napříč webem. Z tohoto důvodu je celkové hodnocení výstupní kvality do určité míry zjednodušené.

### **Tématika webu zkreslující hodnocení vizuálního obsahu**

Zvolená tematika webu, která se soustředí především na technické vybavení, výrazně ovlivnila nároky na vizuální obsah. Vzhledem k nutnosti pracovat s realistickými zobrazeními byly u generovaných obrázků kladený vysoké nároky na přesnost, kontextovou správnost a technickou věrnost. Je pravděpodobné, že v případě abstraktnějšího nebo méně technicky specifického zaměření webu, např. u portfolia fotografa, by výsledky generativní AI dosahovaly lepšího hodnocení, zejména v oblasti estetiky.

### **Míra zkoumání využití AI ve vývoji webů**

Práce by mohla jít více do hloubky v popisu současných přístupů a možností využití umělé inteligence při vývoji webových stránek. Ačkoli se teoretická část zaměřuje na základní principy webového vývoje a obecné trendy v oblasti AI, chybí detailnější rozbor konkrétních metod nebo postupů, které jsou v praxi využívány. Tím by bylo možné doplnit práci o širší kontext a zvýšit její informační hodnotu.

### **Možnost budoucího rozšíření**

Možnosti budoucího rozšíření této práce zahrnují například:

- hlubší porovnání se zahrnutím dalších nástrojů typu NoCode/LowCode, které by mohly být zařazeny jako samostatná kategorie vývoje vedle CMS systémů;
- rozšíření škály vizuálních prvků o ilustrace, animace či videa a sledování, jak si AI poradí s tvorbou multimediálního obsahu napříč různými typy webových prezentací;
- automatizované testování bezpečnosti nebo přístupnosti webů;
- evaluace výsledků pomocí dotazníkového šetření mezi reálnými uživateli webu, čímž by se posílila validita subjektivních aspektů, jako je estetika či relevance obsahu;
- rozšíření teoretické části práce o hlubší rozbor využití AI pro vývoj webů.

# Použitá literatura

- Anttonen, M., Salminen, A., Mikkonen, T., & Taivalsaari, A. (2011). Transforming the web into a real application platform: New technologies, emerging trends and missing pieces. *Proceedings of the 2011 ACM Symposium on Applied Computing*, 800–807. <https://doi.org/10.1145/1982185.1982357>
- Attri, V. K., & Pathania, P. (2025). Challenges of web development. *14, 2025*. <https://doi.org/10.30574/wjaets.2025.14.1.0029>
- Barning, M. (2013, červenec 17). Divide the Web Timeline in nine epochs | Internet with a Brain [Blog]. *Internet with a Brain*. <https://www.web3.lu/divide-the-web-timeline-in-nine-epochs/>
- Goel, A., Bakshi, R., & Krishna, A. (2022). *Web 3.0 and Decentralized Applications*. 2022. <https://doi.org/10.3390/materproc2022010008>
- Jadhav, S. S., & Gholve, S. S. (2024). *The Impact of AI on Web Development*. 3(8). <https://doi.org/10.59828/ijsrnstm.v3i8.240>
- Jazayeri, M. (2007). Some Trends in Web Application Development. *Future of Software Engineering (FOSE '07)*, 199–213. <https://doi.org/10.1109/FOSE.2007.26>
- Jurafsky, D., & Martin, J. H. (2025). *Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models* (3. vyd.).
- Laplante, P. A., & Kassab, M. (2022). *What Every Engineer Should Know about Software Engineering* (2nd Edition). CRC Press.
- Liang, J. T., Yang, C., & Myers, B. A. (2023). *Understanding the Usability of AI Programming Assistants* (arXiv:2303.17125; Verze 1). arXiv. <https://doi.org/10.48550/arXiv.2303.17125>
- Manning, C. (2020, září 1). *Artificial Intelligence Definitions*. Stanford University. <https://hai-production.s3.amazonaws.com/files/2020-09/AI-Definitions-HAI.pdf>

- Mikkonen, T., & Taivalsaari, A. (2010). The mashware challenge: Bridging the gap between web development and software engineering. *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, 245–250. <https://doi.org/10.1145/1882362.1882413>
- Naveed, H., Khan, A. U., Qiu, S., & Saqib, M. (2024). *A Comprehensive Overview of Large Language Models*. <https://doi.org/10.48550/arXiv.2307.06435>
- Purewal, S. (2014). *Learning Web App Development: Build Quickly with Proven JavaScript Techniques* (1st edition). O'Reilly Media.
- Rahaman, A., Gayatri, V., & Kiran, Ch. S. (2022). Development of Web Applications by Integrating Frontend and Backend Tools. *1, 2022*(12).
- Ritesh, Yadav, A., Dubey, A., Prajapati, D., & Srivastava, A. (2023). Evolution of Web Development Frameworks. *International Journal for Research in Applied Science and Engineering Technology*, *11*(11), 2267–2281. <https://doi.org/10.22214/ijraset.2023.57011>
- Rokis, K., & Kirikova, M. (2023). *Exploring Low Code Development: A Comprehensive Literature Review*. *36*. <https://doi.org/10.7250/csimg.2023-36.04>
- Rustambek, M. (2023). THE ROLE OF ARTIFICIAL INTELLIGENCE IN WEB SITES. *Proceedings of International Conference on Scientific Research in Natural and Social Sciences*, *2*(7), Article 7.
- Stocco, A. (2019). How artificial intelligence can improve web development and testing. *Proceedings of the Conference Companion of the 3rd International Conference on Art, Science, and Engineering of Programming*, 1–4. <https://doi.org/10.1145/3328433.3328447>
- Taivalsaari, A., & Mikkonen, T. (2011). The Web as an Application Platform: The Saga Continues. *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*, 170–174. <https://doi.org/10.1109/SEAA.2011.35>
- Upadhyaya, N. (2024). *Artificial Intelligence in Web Development: Enhancing Automation, Personalization, and Decision-Making*. *4*(1).

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need* (arXiv:1706.03762). arXiv.  
<https://doi.org/10.48550/arXiv.1706.03762>
- Xiao, T., & Zhu, J. (2025). *Foundations of Large Language Models*.  
<https://doi.org/10.48550/arXiv.2501.09223>
- Yermolenko, A., & Golchevskiy, Y. (2021). Developing Web Content Management Systems – from the Past to the Future. *SHS Web Conf.*, 110.  
<https://doi.org/10.1051/shsconf/202111005007>

# Přílohy

## Příloha A: Odkazy na webové stránky

CMS řešení: <https://pilazachotin.cz/>

No AI řešení: <https://blackgaben.github.io/pila-zachotin.github.io/>

AI řešení: <https://blackgaben.github.io/pilazachotinai.github.io/>

## Příloha B: Odkazy na GitHub repositáře

No AI řešení: <https://github.com/BlackGaben/pila-zachotin.github.io>

AI řešení: <https://github.com/BlackGaben/pilazachotinai.github.io>