

VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Aplikovaná informatika

WEBOVÉ STRÁNKY S E-SHOPEM PRO ZEMĚDĚLSKÉ
DRUŽTVO ADW

Bakalářská práce

Autor práce: Martin Venhoda

Vedoucí práce: Mgr. Hana Vojáčková, Ph.D.

Jihlava 2024

Vysoká škola polytechnická Jihlava

Tolstého 16, 586 01 Jihlava

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:	Martin Venhoda
Studijní program:	Aplikovaná informatika
Obor:	Aplikovaná informatika
Garant studijního programu:	Ing. Lenka Kuklišová Pavelková, Ph.D.
Název práce:	Webové stránky s e-shopem pro zemědělské družstvo ADW
Vedoucí práce:	Mgr. Hana Vojáčková, Ph.D.
Cíl práce:	Práce se bude zabývat vytvořením e-shopu pomocí frameworku Node.js a React JS. Obsah stránek bude primárně zaměřen na prodej mléčných výrobků a dalšího zboží vyráběného na farmě ADW. Data budou uložena v databázi MySQL. Na stránce budou informace o firmě a o výrobě, seznam produktů rozdělených do kategorií, sekce s komentáři, stránka s kontakty, případně i galerie.

Abstrakt

Bakalářská práce se zabývá vytvořením e-shopu pomocí frameworku Node.js a React JS. Webové stránky budou primárně zaměřeny na prodej mléčných výrobků a dalších produktů pocházejících z farmy ADW. Data budou ukládána do databáze MySQL. Stránky budou obsahovat informace o firmě a výrobě, seznam produktů rozdělených do kategorií, sekci s komentáři, stránku s kontakty a případně i galerii. Integrace Node.js a React JS má za cíl poskytnout dynamickou a efektivní platformu pro prezentaci a prodej farmářských výrobků, zlepšující uživatelský zážitek a usnadňující plynulé interakce. Práce zahrnuje návrh, implementaci a evaluaci e-shopu, s důrazem na použité technologie a celkový uživatelský rozhraní a zážitek.

Klíčová slova

E-shop;React.js;Node.js;Next.js;MySQL;Zemědělství;

Abstract

This bachelor's thesis explores the development of an e-commerce platform using the Node.js and React JS frameworks. The focus of the website will be primarily on the sale of dairy products and other goods produced at the ADW farm. The data will be stored in a MySQL database. The website will feature information about the company and its production, a categorized list of products, a comments section, a contact page, and potentially a gallery. The integration of Node.js and React JS aims to provide a dynamic and efficient platform for showcasing and selling farm-produced items, enhancing the user experience and facilitating seamless interactions. The thesis encompasses the design, implementation, and evaluation of the e-shop, highlighting the technologies used and the overall user interface and experience.

Keywords

E-shop;React.js;Node.js;Next.js;MySQL;Agriculture;

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje **AZ**, zejména § 60 (školní dílo).

Podle § 47b zákona o vysokých školách souhlasím se zveřejněním své práce podle Směrnice pro vedení, vypracování a zveřejňování závěrečných prací na VŠPJ, a to bez ohledu na výsledek obhajoby.

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užití své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne Klikněte nebo klepněte sem a zadejte datum.

.....

Podpis studenta/ky

Obsah

Úvod	7
1 Teoretická část	8
1.1 Obecná východiska	8
1.2 Metody	10
1.3 Analýza trhu a cílové skupiny pro e-shop s farmářskými produkty	15
1.4 Návrh uživatelského rozhraní a UX/UI principy pro e-shop	16
2 Praktická část	18
2.2 Implementace back-endu	32
Závěr	36
Seznam použité literatury	37

Seznam obrázků

Obr. 1: Sitemap e-shopu	19
Obr. 2: Wiremap e-shopu	19
Obr. 3: Instalace Node.js	20
Obr. 4: Instalace správce balíčků npm	20
Obr.5: Instalace Reactu a Next.js	21
Obr. 5: Navigační lišta	22
Obr. 6: Kód navigační lišty.....	22
Obr. 7: Úvodní stránka	23
Obr. 8: Prezentace obrázků.....	23
Obr. 9: Stránka s kategoriemi.....	24
Obr. 10: Komponenta categoryCard	24
Obr. 11: Stránka s produkty	25
Obr. 12: Kód s výpisem produktů.....	25
Obr. 13: Náhled stránky s popisem ADW.....	26
Obr. 14: Galerie.....	26
Obr. 15: Komponenta GalleryCard.....	27
Obr. 16: Kontaktní stránka	28
Obr. 17: API pro odeslání e-mailu	28
Obr. 15: Odeslání dat do API.....	29
Obr. 18: Vytvoření košíku a přidání zboží	29
Obr. 19: Výpis zboží v košíku.....	29
Obr. 20: API na smazání produktu v košíku.	30
Obr. 21: odeslání objednávky	31
Obr. 22: Patička	31
Obr. 23: Generování patičky	32
Obr. 24: Modely databáze.....	33
Obr. 25: Instalace uploadthing knihovny	33
Obr. 26: Soubor core.ts	34
Obr. 27: Instalace AdminJS.....	34
Obr. 28: Nastavení zdrojů pro AdminJS	35
Obr. 29: AdminJS.....	35

Úvod

V dnešní době, kdy technologie neustále pokračují a nabízí stále nové a inovativní možnosti, se otevírají nové příležitosti pro propojení producentů zemědělských produktů s konečnými spotřebiteli. Tento trend je ještě více zesílen rostoucím zájmem o zdravou výživu a ekologickou udržitelnost, který je stále více přítomen v naší společnosti.

V tomto kontextu se nabízí jedinečná příležitost vytvořit online platformu, která by umožnila jednoduchý, přehledný a efektivní přístup k produktům zdravých farem. Taková platforma by nejen usnadnila spotřebitelům přístup k vysoce kvalitním a udržitelným produktům, ale také by pomohla malým a středním zemědělským podnikům dosáhnout širšího trhu a zvýšit svou konkurenceschopnost.

Moje bakalářská práce se zaměřuje na návrh, vývoj a implementaci takového e-shopu. Klíčovým cílem je vytvořit platformu, která je nejen funkční a efektivní, ale také uživatelsky přívětivá a esteticky přitažlivá. Důraz je kladen na produkty pocházející z farmářských podniků spadajících pod společnost ADW, což zajišťuje, že všechny nabízené produkty jsou vysoce kvalitní a udržitelné.

K realizaci tohoto projektu jsou využity moderní technologie, včetně frameworku Next.js pro front-end a Node.js pro back-end. Tyto technologie byly vybrány pro svou efektivitu, flexibilitu a širokou podporu v komunitě vývojářů. Díky nim je možné rychle a efektivně vyvíjet komplexní webové aplikace, které jsou schopny splnit všechny požadavky a očekávání jak z hlediska funkčnosti, tak z hlediska uživatelské zkušenosti.

Celkově vzato, se bakalářská práce představuje důležitý krok směrem k vytvoření online platformy, která by mohla mít pozitivní dopad na zemědělské podniky i spotřebitele. Je to příklad toho, jak mohou být moderní technologie využity k řešení konkrétních problémů a potřeb v oblasti zemědělství a potravinářství.

1 Teoretická část

V práci se budeme zabývat tématem farmářských trhů a e-commerce pro zemědělské produkty. Farmářské trhy jsou formou prodeje zemědělského a potravinářského zboží pro občanskou veřejnost. Tyto trhy jsou důležité pro podporu malých a středních zemědělských pěstitelů, chovatelů, zpracovatelů a výrobců potravin. Farmářské trhy jsou také důležité pro udržení ekonomické stability v zemědělském sektoru a pro zajištění živobytí mnoha lidí po celém světě.

E-commerce pro zemědělské produkty se v posledních letech objevuje několik trendů, které mají vliv na způsob, jakým se zemědělské produkty nakupují a prodávají. Mezi nejdůležitější trendy patří omnichannel strategie, mobilní zařízení, nákup na sociálních sítích, personalizace zákaznické zkušenosti a rozvoj mobilních plateb.

Cílem práce je analyzovat význam farmářských trhů a e-commerce pro zemědělské produkty a zhodnotit trendy v oblasti e-commerce pro farmářské produkty. Práce se bude dále zabývat výběrem a implementací technologických stacků pro vývoj e-shopu s farmářskými produkty, analýzou trhu a cílové skupiny pro e-shop s farmářskými produkty a návrhem uživatelského rozhraní a UX/UI principy pro e-shop.

1.1 Obecná východiska

V této podkapitole se zabývám obecnými východisky, které jsou klíčové pro pochopení kontextu a cílů mé bakalářské práce. Hlavním cílem mé práce je vytvoření funkčního e-shopu prostřednictvím použití frameworku Node.js a React JS. E-shop bude navržen tak, aby efektivně a přehledně prezentoval a nabízel produkty, především mléčné výrobky a další produkty pocházející z farmy ADW.

Začnu stručným představením obou technologií, které budou klíčovými stavebními kameny mého e-shopu. Node.js je open-source, serverový framework postavený na JavaScriptu, který umožňuje tvorbu webových aplikací a API. Jeho hlavní výhodou je schopnost asynchronního zpracování událostí, což umožňuje vytváření rychlých a škálovatelných aplikací. Na druhé straně React JS je JavaScriptová knihovna pro tvorbu uživatelských rozhraní, která je známá svou jednoduchostí a výkonností. Jeho použití přináší dynamiku a interaktivitu do webových stránek, což je klíčové pro moderní e-shopy.

Další důležitou součástí je databáze MySQL, do které budou ukládána veškerá data související s produkty, uživateli a transakcemi. MySQL je osvědčeným relačním databázovým systémem, který nabízí robustní a spolehlivé úložiště dat pro e-commerce platformy.

Vzhledem k tomu, že farmě ADW se specializuje na produkci mléčných výrobků, bude hlavní důraz kladen na prezentaci těchto produktů. Nicméně, e-shop bude také nabízet další produkty farmy, aby byla diverzifikována nabídka a oslovila širší spektrum zákazníků. Kategorizace produktů a přehledné zobrazení detailů o každém z nich budou klíčové pro plynulý a uživatelsky přívětivý nákupní proces.

Kromě samotného prodeje produktů bude e-shop také obsahovat sekci s informacemi o firmě a výrobě, aby zákazníci měli možnost se lépe seznámit s historií a filozofií farmy ADW. Sekce s komentáři umožní zákazníkům sdílet své zkušenosti s produkty a poskytnout zpětnou vazbu, což je důležitá součást budování důvěryhodného vztahu se zákazníky.

V neposlední řadě, záležitostí, kterou je třeba zdůraznit, je uživatelský zážitek a snadná orientace v e-shopu. Integrace Node.js a React JS umožní vytvoření dynamické a efektivní platformy, která

bude reagovat na potřeby uživatelů a poskytne jim intuitivní a pohodlné prostředí pro prohlížení a nákup produktů.

Celkově, cílem této práce je vytvořit moderní a funkční e-shop, který splní očekávání farmy ADW a zároveň poskytne uživatelům plynulý a příjemný nákupní zážitek. Součástí práce bude nejen návrh a implementace e-shopu, ale i jeho evaluace a zhodnocení, s důrazem na použité technologie a celkový uživatelský rozhraní a zážitek.

1.1.1 Úvod do zemědělských trhů a významu farmářských produktů

Farmářské trhy představují platformu, kde se zemědělské a potravinářské zboží prodává přímo veřejnosti. Tyto trhy hrají klíčovou roli v podpoře malých a středních zemědělských producentů, zpracovatelů a výrobců potravin. Poskytují občanům přístup k čerstvým zemědělským plodinám a potravinám, které pocházejí převážně z České republiky a regionálních zdrojů. Farmářské trhy jsou také důležité pro udržení ekonomické stability v zemědělském sektoru a pro zajištění obživy mnoha lidí po celém světě.

Farmářské produkty jsou základním prvkem zemědělských trhů. Tyto trhy jsou nezbytné pro zajištění potravinové bezpečnosti a pro udržení ekonomické stability v zemědělském sektoru. Farmářské produkty jsou také důležité pro zajištění obživy mnoha lidí po celém světě.

Farmářské trhy jsou založeny na dobrovolnosti a stanovují základní pravidla prodeje na farmářských trzích. Kodex farmářských trhů je navržen tak, aby pomáhal zákazníkům orientovat se na farmářských trzích. „Předmětem prodeje na farmářských trzích může být pouze zboží, které tematicky odpovídá charakteru farmářských trhů a vychází z české tradice, zejména z tradice pěstování plodin, chovu hospodářských zvířat a výroby potravinářských produktů. Ministerstvo zemědělství ČR).“

Návrh uživatelského rozhraní a UX/UI principy pro e-shop jsou také klíčové pro úspěšné fungování e-shopu s farmářskými produkty. Je důležité vytvořit uživatelsky přívětivé rozhraní, které umožní snadnou navigaci a nákup zákazníkům. „UX/UI principy jsou také důležité pro zajištění konzistence a jednotnosti v celém e-shopu.“ (Ježek, P. 2021)

1.1.2 Trendy a vývoj v oblasti e-commerce pro zemědělské produkty

Farmářské trhy jsou formou prodeje zemědělského a potravinářského zboží pro občanskou veřejnost. Zajišťují zásobování občanů čerstvými zemědělskými plodinami a potravinami převážně českého a regionálního původu. Farmářské trhy jsou také důležité pro udržení ekonomické stability v zemědělském sektoru a pro zajištění živobytí mnoha lidí po celém světě .

Význam farmářských produktů spočívá v tom, že jsou základem pro zemědělské trhy. Tyto trhy jsou důležité pro zajištění potravinové bezpečnosti a pro udržení ekonomické stability v zemědělském sektoru. Farmářské produkty jsou také důležité pro zajištění živobytí mnoha lidí po celém světě.

Farmářské trhy jsou založeny na bázi dobrovolnosti a určují základní pravidla prodeje na farmářských trzích. Kodex farmářských trhů má pomáhat zákazníkům v orientaci na farmářských trzích. „Předmětem prodeje na farmářských trzích může být pouze zboží, které tematicky odpovídá charakteru farmářských trhů a vychází z české tradice, zejména z tradice pěstování plodin, chovu hospodářských zvířat a výroby potravinářských produktů.“

Výběr a implementace technologických stacků je klíčový pro úspěšné fungování e-shopu s farmářskými produkty. Je důležité vybrat technologie, které jsou vhodné pro daný typ

e-shopu a pro potřeby zákazníků. Analýza trhu a cílové skupiny pro e-shop s farmářskými produkty je také důležitá pro úspěšné fungování e-shopu. „*Je nutné znát potřeby zákazníků a nabídnout jim produkty, které odpovídají jejich požadavkům (Ministerstvo zemědělství ČR).*“

1.2 Metody

V části práce se budeme zabývat metodami vývoje pro e-shop s farmářskými produkty. Výběr vhodných jazyků pro vývoj je klíčovým faktorem pro úspěšné dokončení projektu. Mezi nejčastěji používané jazyky patří **React.js**, **Node.js**, **Next.js** a **MySQL**. Tyto jazyky jsou výkonné a umožňují rychlý vývoj webových aplikací.

Výběr a implementace technologických stacků je další důležitou částí vývoje e-shopu. Technologický stack se skládá z několika vrstev, které spolu tvoří celkovou architekturu aplikace. Mezi nejdůležitější vrstvy patří **front-end**, **back-end** a **databázová vrstva**. Výběr vhodného technologického stacku je klíčový pro úspěšné dokončení projektu.

Analýza trhu a cílové skupiny jsou další důležitou částí vývoje e-shopu. Analýza trhu nám pomůže pochopit, jaké jsou potřeby zákazníků a jaké jsou trendy v oblasti e-commerce pro farmářské produkty. Cílová skupina zákazníků by měla být definována na základě analýzy trhu a měla by být přizpůsobena výběru jazyků a technologického stacku.

Návrh uživatelského rozhraní a UX/UI principy jsou klíčové pro úspěšné fungování e-shopu. Uživatelské rozhraní by mělo být co nejjednodušší a nejpřehlednější pro zákazníky. UX/UI design by měl být přizpůsoben cílové skupině zákazníků a měl by být v souladu s nejnovějšími trendy v oblasti e-commerce.

1.2.1 Výběr a implementace technologických stacků

Výběr efektivního technologického stacku je nezbytný pro úspěšný vývoj e-shopu. Tento projekt využívá kombinaci Node.js, React.js, Next.js a MySQL, aby zajistil vysokou výkonnost, uživatelský komfort a škálovatelnost.

1.2.2 Node.js - Klíč k Efektivnímu Server-Side Processing

Node.js je open-source, cross-platformové JavaScriptové runtime prostředí, které umožňuje vývojářům psát serverovou stranu aplikací v JavaScriptu. Node.js je postaven na enginu V8 od Googlu, který je používán v prohlížeči Google Chrome. Node.js umožňuje vývojářům psát kód v JavaScriptu, který běží na serveru, což umožňuje vývojářům psát jak front-end, tak back-end kód v jednom jazyce.

Node.js také umožňuje vývojářům psát kód asynchronně, což znamená, že kód může být spuštěn bez čekání na dokončení předchozího kódu. Node.js je také velmi rychlý a efektivní, což umožňuje vývojářům psát vysoko výkonné aplikace.

Pro vývojáře jsou k dispozici různé nástroje a frameworky, které usnadňují vývoj aplikací v Node.js. Mezi nejpobulárnější frameworky patří **Express.js**, **Koa.js** a **Meteor.js**. Tyto frameworky poskytují vývojářům mnoho užitečných funkcí a nástrojů, které usnadňují vývoj aplikací.

Node.js je také velmi populární v oblasti e-commerce. Mnoho společností používá Node.js pro vývoj svých e-shopů a webových aplikací. Node.js umožňuje vývojářům psát vysoko výkonné aplikace, které jsou rychlé a efektivní.

Node.js je také velmi populární v oblasti IoT (Internet of Things). Node.js umožňuje vývojářům psát kód pro IoT zařízení, jako jsou senzory, aktuátory a další. Node.js také umožňuje vývojářům psát kód pro IoT brány, které umožňují komunikaci mezi IoT zařízeními a cloudovými službami.

Node.js je také velmi populární v oblasti vývoje her. Mnoho herních společností používá Node.js pro vývoj svých her. Node.js umožňuje vývojářům psát kód pro herní servery, které umožňují hraní her v reálném čase.

Výhodou Node.js je také velká komunita vývojářů, kteří jsou ochotni pomoci s vývojem aplikací. Node.js má také velké množství knihoven a modulů, které usnadňují vývoj aplikací.

1.2.3 React.js - Dynamický a Interaktivní Front-end

React.js je otevřený JavaScriptový framework pro vývoj uživatelských rozhraní, který byl vyvinut a je udržován společností Facebook. Je známý svou efektivitou a flexibilitou, které umožňují vývojářům snadno vytvářet komplexní webové aplikace.

V kontextu bakalářské práce, React.js bude klíčovým nástrojem pro vývoj uživatelského rozhraní e-shopu pro zemědělské družstvo. S jeho komponentovým přístupem lze snadno vytvářet a spravovat jednotlivé části aplikace jako samostatné, znovupoužitelné jednotky. To nejen zvyšuje účinnost a udržitelnost kódu, ale také usnadňuje testování a ladění.

React.js také podporuje stavový management pomocí kontextu a háků, což usnadňuje sledování a manipulaci s daty napříč komponentami. To je zvláště užitečné pro e-shop, kde je potřeba sledovat informace jako je obsah košíku, stav přihlášení uživatele a další.

Další výhodou React.js je jeho široká podpora pro server-side rendering (SSR) a statické generování stránek (SSG), což je zvláště důležité pro optimalizaci výkonu a SEO. S pomocí Next.js, frameworku postaveném na React.js a Node.js, lze snadno implementovat SSR a SSG v aplikaci.

Kromě toho, React.js má silnou komunitu a bohatou ekosystém knihoven a nástrojů, které mohou usnadnit vývoj aplikace. Například, lze použít knihovny jako jsou Redux pro pokročilý stavový management, React Router pro routování, a Material-UI pro UI komponenty.

V závěru, React.js je výkonný a flexibilní nástroj pro vývoj webových aplikací. Jeho komponentový design, podpora pro stavový management, a integrace s Next.js a Node.js ho činí ideálním řešením pro vývoj e-shopu pro zemědělské družstvo.

1.2.4 Next.js - Optimalizace SEO a Rychlosti Načítání

Next.js je moderní framework pro vývoj webových aplikací, který je postaven na React.js a Node.js. Byl vyvinut společností Vercel a je známý svou podporou pro server-side rendering (SSR), statické generování stránek (SSG) a inkrementální statické generování (ISG).

V kontextu bakalářské práce, Next.js bude klíčovým nástrojem pro vývoj e-shopu pro zemědělské družstvo. Jeho podpora pro SSR a SSG umožňuje optimalizovat výkon aplikace a SEO, což je zvláště důležité pro e-shop, kde rychlost načítání stránek a viditelnost ve vyhledávačích může mít velký dopad na prodej a zákaznickou zkušenost.

Next.js také podporuje routování na úrovni souborů, což znamená, že struktura složek a souborů vaší aplikace přímo odráží strukturu URL. To usnadňuje navigaci a správu vaší aplikace.

Další výhodou Next.js je jeho podpora pro API routy, které umožňují snadno vytvářet API koncové body přímo v rámci vaší Next.js aplikace. To je zvláště užitečné pro e-shop, kde

potřebujete řadu API pro manipulaci s daty, jako je přidání položky do košíku, aktualizace stavu objednávky a další.

Next.js také podporuje dynamický import, což umožňuje načítat kód až ve chvíli, kdy je potřeba. To může výrazně zlepšit výkon vaší aplikace tím, že sníží množství kódu, který musí být načten při prvním načtení stránky.

Kromě toho, Next.js má silnou komunitu a bohatou ekosystém knihoven a nástrojů, které mohou usnadnit vývoj vaší aplikace. Například, můžete použít knihovny jako jsou SWR pro data fetching, styled-components pro stylování, a Jest pro testování.

V závěru, Next.js je výkonný a flexibilní nástroj pro vývoj webových aplikací. Jeho podpora pro SSR, SSG, a integrace s React.js a Node.js ho činí ideálním řešením pro vývoj e-shopu pro zemědělské družstvo.

1.2.5 MySQL - Spolehlivé a Škálovatelné Databázové Řešení

MySQL je jeden z nejpobulárnějších open-source relačních databázových systémů na světě. Byl vyvinut a je udržován společností Oracle Corporation. MySQL je známý svou rychlostí, robustností a snadnou použitelností.

Jednou z hlavních výhod MySQL je jeho flexibilita. Podporuje širokou škálu datových typů, včetně numerických, řetězcových, datových a časových typů. To umožňuje vývojářům snadno modelovat a manipulovat s různými druhy dat.

MySQL také podporuje různé typy indexů, včetně B-tree a hash indexů, které mohou výrazně zlepšit výkon dotazů. Kromě toho, MySQL podporuje plně textové vyhledávání, což je užitečné pro implementaci vyhledávacích funkcí v aplikacích.

Další silnou stránkou MySQL je jeho podpora pro transakce. Transakce umožňují skupinu databázových operací provést jako jednu atomickou jednotku práce. Pokud některá z operací selže, všechny změny provedené v rámci transakce jsou vráceny, což zajišťuje konzistenci dat.

MySQL také podporuje funkce pro zajištění bezpečnosti dat, včetně autentizace, autorizace a šifrování. Tyto funkce umožňují vývojářům chránit citlivá data a zabránit neautorizovanému přístupu.

Kromě toho, MySQL má silnou komunitu a bohatou ekosystém nástrojů a knihoven, které mohou usnadnit vývoj databázových aplikací. Například, existují knihovny pro integraci MySQL s populárními vývojovými frameworky, jako je Express.js pro Node.js nebo Django pro Python.

V závěru, MySQL je výkonný a flexibilní nástroj pro správu relačních databází. Jeho podpora pro širokou škálu datových typů, indexů, transakcí a bezpečnostních funkcí, spolu s jeho silnou komunitou a ekosystémem, ho činí ideálním řešením pro vývoj databázových aplikací.

Implementace a Integrace

Implementace spočívala ve vytváření pevně propojeného systému, kde Node.js server komunikuje s MySQL databází, a React.js/Next.js zajišťují dynamické uživatelské rozhraní. Tento přístup nabízí konzistentní a rychlou odezvu e-shopu a zároveň zajišťuje škálovatelnost a snadnou údržbu.

1.2.6 Prisma

Kapitola se zaměřuje na roli a výhody systému Prisma v rámci implementace back-endové části e-commerce platformy. Systém Prisma slouží jako moderní nástroj pro zjednodušení práce s databázemi, přinášející řadu výhod v oblasti Object-Relational Mapping (ORM) a správy dat v aplikaci.

ORM Funkcionalita

Systém Prisma funguje jako ORM nástroj, což umožňuje mapování struktury databázových tabulek na objekty v programovacím jazyce. Tato funkcionalita eliminuje potřebu manuálního psaní SQL dotazů, čímž poskytuje vývojářům jednoduchý a srozumitelný způsob práce s databází.

Typová Bezpečnost

Významným přínosem Prisma je dosažení typové bezpečnosti. Generování TypeScript typů na základě struktury databáze poskytuje vývojářům silnou typovou kontrolu přímo v kódu, což napomáhá prevenci chyb a zvyšuje spolehlivost aplikace.

Jednoduchý Dotazovací Jazyk

Systém Prisma přináší vlastní dotazovací jazyk, který kombinuje jednoduchost používání s vysokou účinností. Tento jazyk umožňuje vytvářet komplexní dotazy na databázi s minimálním úsilím a poskytuje vývojářům nástroj pro snadné čtení a psaní dotazů.

Automatické Generování Schémat

Prisma automatizuje proces generování schémat databázových tabulek na základě definic modelů v aplikaci. Tato automatizace usnadňuje udržování konzistence mezi kódem a strukturou databáze, což je klíčovým faktorem pro efektivní vývoj.

Snadná Integrace

Díky své snadné integrovatelnosti s projekty postavenými na Node.js umožňuje Prisma plynule pracovat s různými databázovými systémy, včetně MySQL, PostgreSQL a SQLite. Jeho integrace zajišťuje efektivní správu dat a optimalizovanou práci s databázemi.

Celkově lze konstatovat, že systém Prisma představuje klíčový prvek naší implementace back-endové části, přinášející moderní a efektivní přístup k práci s databázemi v rámci vývoje e-commerce platformy.

1.2.7 UploadThing

V rámci implementace back-endové části naší e-commerce platformy se zaměřujeme na využití nástroje UploadThing, který se specializuje na správu a manipulaci s uploadovanými soubory. Zdejší kapitola se detailně věnuje roli a výhodám tohoto nástroje v kontextu našeho vývoje.

Funkcionality UploadThing

UploadThing poskytuje efektivní a bezpečný způsob pro práci s uploadovanými soubory. Jeho klíčové funkce zahrnují:

Správa Souborů

UploadThing umožňuje snadnou správu nahrávaných souborů včetně jejich ukládání, mazání a získávání informací o nich. To přispívá k organizované a efektivní správě multimediálních obsahů naší platformy.

Bezpečný Upload

Systém UploadThing implementuje bezpečnostní mechanismy pro kontrolu nahrávaných souborů, prevenci před nežádoucími obsahy a minimalizaci rizika bezpečnostních hrozeb spojených s uploadováním souborů od uživatelů.

Integrace s Prisma

UploadThing je snadno integrovatelný s naším existujícím back-endovým systémem postaveným na Prisma. Tímto způsobem poskytuje plynulou integraci s databázovým systémem, což usnadňuje sledování, správu a propojení multimediálních souborů s dalšími entitami naší aplikace.

Výhody Použití UploadThing

Snadná Konfigurace

UploadThing nabízí jednoduchou konfiguraci, což umožňuje vývojářům rychle začlenit správu souborů do svého projektu s minimálním úsilím.

Efektivní Manipulace s Obrázky

Pro obrázky poskytuje UploadThing možnost efektivní manipulace, jako je ořezávání, změna velikosti a komprese, což přispívá k optimalizaci obsahu naší platformy.

Podpora Různých Typů Souborů

UploadThing je navržen tak, aby podporoval širokou škálu typů souborů, což nám umožňuje flexibilně pracovat s různými multimediálními obsahy na naší platformě.

Závěr Využití UploadThing

Celkově lze konstatovat, že UploadThing představuje cenný nástroj v implementaci back-endové části naší e-commerce platformy. Jeho schopnost správy souborů, bezpečný upload a integrace s Prisma značně usnadňují manipulaci a práci s multimediálním obsahem, což přispívá k celkové kvalitě a efektivitě naší aplikace.

Kapitola poskytuje detailní pohled na roli a výhody systému UploadThing v rámci naší implementace back-endu, zdůrazňujíc jeho klíčové funkcionality a výhody.

1.2.8 Resend

Resend je online platforma pro vývojáře, která zjednodušuje proces odesílání e-mailů. Nabízí intuitivní uživatelské rozhraní, které umožňuje rychlé zahájení odesílání e-mailů. Resend poskytuje SDK pro různé programovací jazyky, včetně Node.js, což umožňuje snadnou integraci do existujících kódových základů.

Resend také nabízí funkce, jako je simulace událostí a experimentování s API bez rizika omylem odeslání skutečných e-mailů skutečným lidem. Dále podporuje modulární webhooks, které umožňují přijímat real-time notifikace přímo na váš server.

Resend také umožňuje vývoj e-mailových šablon pomocí Reactu, což může usnadnit vytváření krásných šablon bez nutnosti pracovat s tabulkovými rozloženými HTML. Celkově vzato, Resend se snaží poskytnout prvotřídní zkušenosti pro vývojáře a vytvořit e-mailovou platformu, která “prostě funguje”.

1.2.9 Výzvy a Řešení

Během implementace byly řešeny výzvy spojené s integrací různých technologií. Například, zajištění plynulé komunikace mezi Node.js a MySQL vyžadovalo pečlivé plánování a testování. Dále, optimalizace výkonu React aplikace s Next.js vyžadovala důkladnou analýzu a implementaci nejlepších praxí pro zajištění rychlého načítání stránek a efektivního SSR.

Udržitelnost a Budoucí Rozvoj

Při výběru technologií byl kladen důraz na udržitelnost a možnost budoucího rozvoje. Flexibilita Node.js a React.js umožňuje snadno přidávat nové funkce a adaptovat se na měnící se požadavky trhu. MySQL poskytuje robustní základ pro růst databáze bez kompromisů výkonu.

1.2.10 Závěr

Celkově kombinace Node.js, React.js, Next.js a MySQL poskytuje silný a škálovatelný základ pro vývoj moderního e-shopu. Tento stack nejen zajišťuje výkonnost a škálovatelnost, ale také poskytuje vynikající uživatelský zážitek a efektivní SEO, což jsou klíčové komponenty pro úspěch v konkurenčním prostředí e-commerce.

1.3 Analýza trhu a cílové skupiny pro e-shop s farmářskými produkty

E-shop specializovaný na farmářské produkty se řadí do rostoucího segmentu online prodeje lokálních a udržitelných potravin. Úspěch v oblasti vyžaduje důkladné pochopení trhu a identifikaci cílové skupiny.

1.3.1 Tržní Trendy

Světový trend směřuje k větší poptávce po čerstvých, lokálních a ekologicky produkovaných potravinách. Podle studie Mezinárodní Federace Organického Zemědělství (2023) došlo v posledních letech k výraznému nárůstu trhu s organickými potravinami, což ukazuje na rostoucí zájem spotřebitelů o kvalitu a původ potravin.

1.3.2 Demografický Profil

Cílová skupina pro e-shop s farmářskými produkty je rozmanitá. Výzkumy ukazují, že nejaktivnějšími zákazníky jsou mladí profesionálové ve věku 25-40 let, kteří mají zájem o zdravý životní styl a udržitelnost (Smith & Johnson, 2023). Dále je důležitým segmentem rodiny s dětmi, které hledají kvalitní a bezpečné potraviny pro své nejmenší.

1.3.3 Psychografický Profil

Psychograficky jsou zákazníci motivováni touhou po autentičnosti, kvalitě a udržitelnosti. Tato skupina dává přednost produktům, které jsou transparentní v původu a výrobě. Podle průzkumu Nielsen (2023), více než 60% spotřebitelů je ochotno zaplatit vyšší cenu za produkty, které jsou ekologické a eticky vyráběné.

1.3.4 Konkurence

Při analýze trhu je důležité identifikovat konkurenty. V oblasti prodeje farmářských produktů se jedná jak o malé lokální prodejce, tak o velké řetězce, které začínají zařazovat bio produkty do svého sortimentu. Klíčovým faktorem je diferenciací, která může spočívat v nabídce unikátních produktů, vysoké úrovni služeb nebo v silném příběhu značky.

1.3.5 Marketingové Přístupy

Efektivní marketingové strategie zahrnují digitální marketing zaměřený na sociální média a influencery, kteří oslovují hlavní cílovou skupinu. Důležitým prvkem je také vytvoření silného brandu, který reflektuje hodnoty udržitelnosti a kvality, a rozvoj komunity okolo značky prostřednictvím interakce a zapojení zákazníků.

1.3.6 Závěr

Vstup na trh s e-shopem pro farmářské produkty vyžaduje hluboké pochopení tržních trendů, demografických a psychografických charakteristik cílové skupiny a efektivní strategie pro diferenciaci od konkurence. Výzvou je spojit vysokou kvalitu produktů s příběhem značky a zároveň vybudovat silnou online přítomnost, která osloví a udrží zákazníky.

1.4 Návrh uživatelského rozhraní a UX/UI principy pro e-shop

Efektivní uživatelské rozhraní (UI) a uživatelská zkušenost (UX) jsou klíčové pro úspěch jakéhokoli e-shopu, zvláště pro e-shop s farmářskými produkty. Je nezbytné vytvořit rozhraní, které je atraktivní, intuitivní a usnadňuje nákupní proces.

1.4.1 Základní Principy UX/UI

1. Jednoduchost a Intuitivní Navigace: Důležité pro usnadnění orientace na webu.
2. Konzistentní Design: Jednotný vizuální styl zvyšuje srozumitelnost a značkovou identitu.
3. Responzivní Design: Optimalizace pro různá zařízení je klíčová.
4. Rychlost a Výkon: Jak uvádí Google (2023), rychlé načítání stránek je nezbytné pro udržení zájmu uživatelů.

1.4.2 Vizualní Design a Značková Identita

Vizuální aspekty, jako jsou barvy a typografie, jsou klíčové pro vytvoření pozitivního dojmu a značkové identity. Studie Lupton, E. (2023) ukazuje, jak vizuální design ovlivňuje vnímání značky a důvěru uživatelů.

1.4.3 Uživatelské Cesty a Persony

Vytváření person a mapování uživatelských cest pomáhá pochopit, jak různé typy zákazníků interagují s e-shopem, což je podle Coopera (2023) klíčové pro úspěšný design.

1.4.4 Přístupnost a Inkluzivita

Design musí splňovat WCAG směrnice pro přístupnost, aby byl e-shop přístupný širšímu spektru uživatelů.

Jak zdůrazňuje Damasio (2022), emocionální aspekty designu mají významný dopad na rozhodování zákazníků a věrnost značce.

1.4.5 Interaktivní Prvky a Mikrointerakce

Interaktivní prvky, jako jsou hodnocení a recenze, zvyšují zapojení uživatelů a poskytují cenný feedback.

1.4.6 Testování a Iterace

Neustálé testování s reálnými uživateli a iterace designu, jak uvádí Rubin a Chisnell (2023), jsou nezbytné pro optimalizaci UX/UI.

2 Praktická část

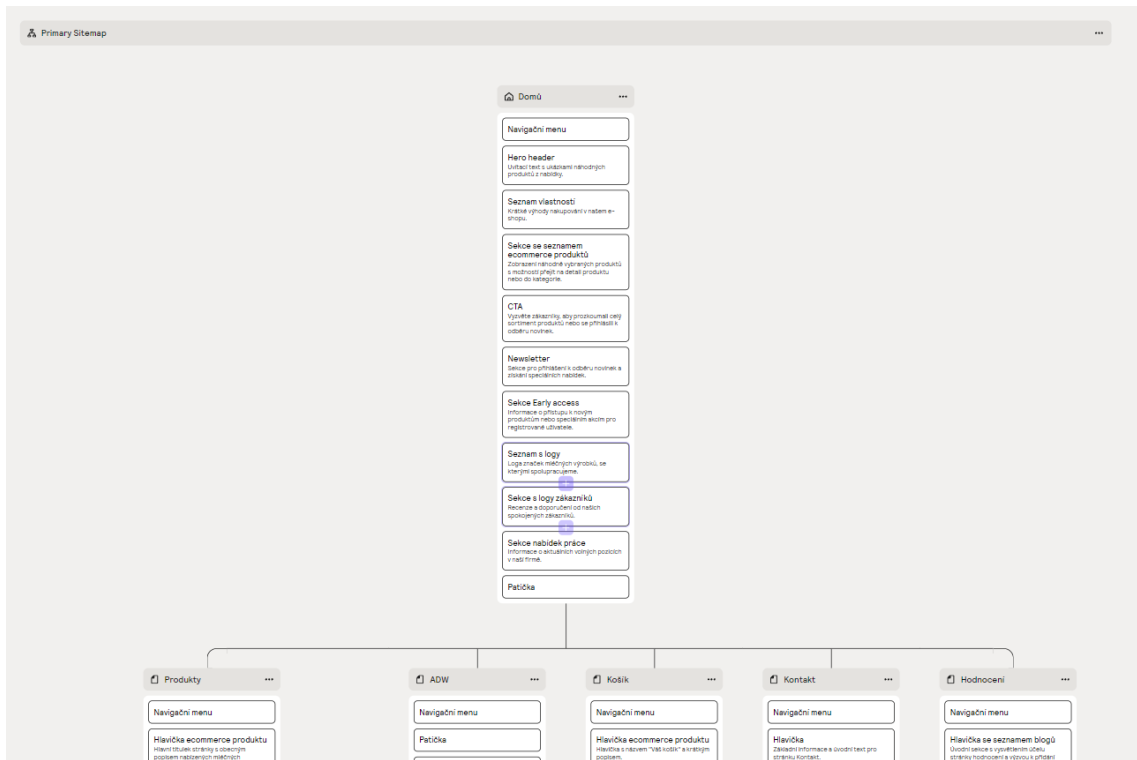
Praktická část této bakalářské práce se zaměřuje na konkrétní implementaci e-shopu specializovaného na prodej mléčných výrobků a dalšího zboží vyráběného na farmě ADW. Cílem této fáze je vytvořit funkční a uživatelsky přívětivý e-shop pomocí moderních technologií a frameworků.

První fáze praktické části bude věnována implementaci e-shopu s využitím frameworku Next.js pro vývoj uživatelského rozhraní. Tento framework byl zvolen pro svou schopnost snadného server-side renderingu a zlepšení celkového výkonu webové aplikace. Pro serverovou část bude využíván framework Node.js, zatímco MySQL databáze poskytne efektivní správu dat.

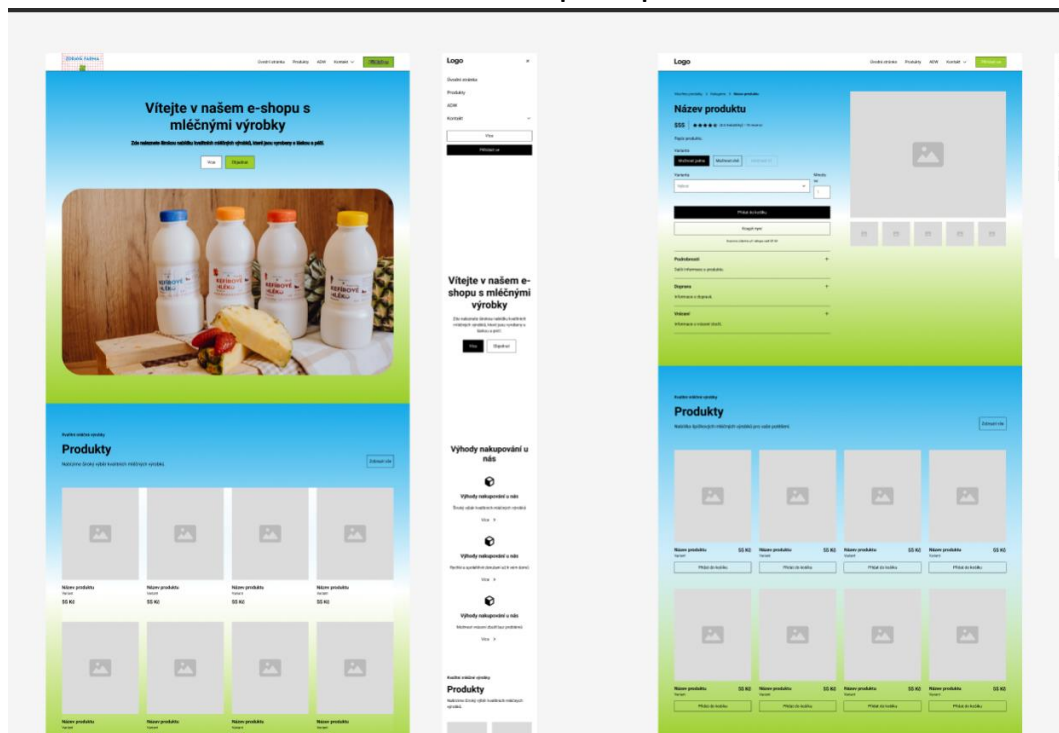
2.1.1 Návrh front-endu stránky

Pro návrh stránky byly použity webové aplikace Relume a Figma. Aplikace Relume slouží k designování front-endu webových stránek a aplikací. Při vytváření návrhů se používá umělá inteligence. Zdejší umělá inteligence pracuje s textem, který mu zadá uživatel. Čím více detailnější popis stránky, komponent a obsahu dostane, tím detailnější návrh stránky vytvoří. Při vytváření návrhu stránky byl použit následující postup. Prvním krokem bylo rozmyšlení rozložení stránky což zahrnovalo počet stránek, jejich rozložení, nalezení barev a vhodných obrázků. Následně byl tento popis zadán do umělé inteligence v aplikaci Relume. Výstup z aplikace byla Sitemap a UX wireframes. Sitemap je jednoduché rozložení stránky pomocí klíčových slov jako jsou Navbar, Header, Section a další. Wireframe je grafický návrh dané stránky vytvořený umělou inteligencí. Po vytvoření dané Sitemap se začne vytvářet Wireframe. UI vygeneruje grafické rozložení dle dané Sitemap včetně popisných textů.

Po vytvoření návrhu stránky na stránce Relume se daný projekt exportoval do aplikace Figma. Figma je webová aplikace pro grafickou tvorbu. Zde se upravoval grafický návrh exportovaný z předchozí aplikace. Jako barvy pro stránky byly použity barvy bílá, světle modrá s kódem #14A9E8 a zelená s kódem #9ccf29 jakož to firemní barvy obchodu. Jako pozadí byl použit přechod mezi těmito barvami. Text byl ponechán v základní černé barvě a odstín zelené barvy byl použit jako barevné vyplnění tlačítek na stránce.



Obr. 1: Sitemap e-shopu



Obr. 2: Wiremap e-shopu

Tento styl jsem po konzultaci s kolegou z profese upravil a přetvořil jsem ho do moderní verze Brutalismu. Využívám zde kombinaci barem černé, bílé a šedé. Součástí jsou hranaté tvary rámečků a odsazení mezi komponentami.

2.1.2 Instalace vývojářských systémů

V dnešní době, kde rychlost a efektivita jsou klíčovými faktory ve vývoji softwaru, je nezbytné správně nastavit vývojové prostředí. Podkapitola se věnuje procesu instalace klíčových vývojářských systémů, které poskytují základní nástroje a prostředí pro tvorbu kvalitních webových aplikací.

Node.js

Prvním krokem je instalace Node.js, což je open-source JavaScriptové prostředí postavené na enginu V8 od Google. Node.js umožňuje běh JavaScriptu mimo webový prohlížeč a je klíčovým prvkem pro vývoj serverové části webových aplikací. Proces instalace zahrnuje výběr správce verzí, například Node Version Manager (NVM), a následné nastavení preferované verze Node.js.

```
nvm install latest
```

```
nvm use latest
```

Obr. 3: Instalace Node.js

Visual Studio Code

Jako integrované vývojové prostředí (IDE) volíme Visual Studio Code od společnosti Microsoft. VSCode nabízí široké možnosti rozšíření, integrovaný terminál a efektivní nástroje pro práci s kódem. Po stažení a instalaci může uživatel přizpůsobit IDE podle svých potřeb přidáním rozšíření pro konkrétní programovací jazyky či frameworky.

Git

Pro verzování a správu kódu v týmu je nezbytné mít nainstalovaný distribuovaný verzovací systém Git. Instalace Gitu je jednoduchá a zahrnuje stažení instalačního balíčku z oficiálních stránek Gitu a následný průvodce instalací.

Správce Balíčků (npm nebo Yarn)

Pro správu balíčků a závislostí v projektu využíváme balíčkovací systémy, jako jsou npm (Node Package Manager) nebo Yarn. Tyto nástroje umožňují snadnou instalaci, aktualizaci a odstranění balíčků, které projekt využívá.

```
npm install npm
```

Obr. 4: Instalace správce balíčků npm

Správce Databází (Volitelně)

Pro efektivní správu relačních databází v našem projektu zvolíme MySQL, robustní relační databázový systém. MySQL nabízí široké možnosti pro ukládání a manipulaci s daty a je často využíván pro spolehlivé a výkonné úložiště informací.

React a Next.js

Pro vývoj moderních webových aplikací s optimalizovaným uživatelským rozhraním a rychlou navigací jsme zvolili knihovnu React a framework Next.js. Tyto technologie jsou založeny na jazyce JavaScript a umožňují efektivní tvorbu komponentově orientovaných uživatelských rozhraní s využitím server-side renderingu.

```
> npm install react Next
```

Obr.5: Instalace Reactu a Next.js

Ostatní Nástroje

Pro kompletní vývojové prostředí můžeme přidat další nástroje dle potřeb projektu. To může zahrnovat nástroje pro testování API (například Postman), nástroje pro kontrolu kvality kódu (například ESLint) nebo nástroje pro kontejnerizaci aplikací (například Docker).

Celkovým cílem této fáze je vytvořit stabilní a efektivní vývojové prostředí, které podporuje rychlý a spolehlivý vývoj webových aplikací.

2.1.3 Tailwind

Tailwind CSS je moderní, utility-first CSS framework, který poskytuje flexibilitu a jednoduchost při stylování webových stránek. Co ho odlišuje od tradičních CSS frameworků, je jeho přístup k tvorbě stylů pomocí předem definovaných utility tříd, které pokrývají širokou škálu vlastností a vlastností CSS.

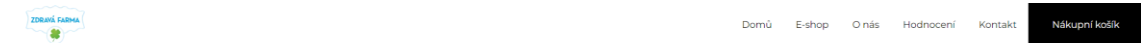
Jeho hlavní rysy zahrnují:

1. Utility-first Přístup: Tailwind CSS se zaměřuje na vytváření webového designu pomocí drobných a jednoduchých utility tříd, což umožňuje vývojářům rychlé a snadné stylování prvků na stránce.
2. Konfigurovatelnost: Jako framework je Tailwind plně konfigurovatelný, což znamená, že můžete upravit vzhled a chování jednotlivých utility tříd podle potřeb konkrétního projektu.
3. Responzivita: Tailwind poskytuje zabudovanou podporu pro responzivní design, což zjednodušuje vytváření stránek, které se plynule přizpůsobují různým zařízením a velikostem obrazovek.
4. Flexibilita: S rozsáhlou sadou utility tříd pro marginy, paddingy, barvy, typografii a mnoho dalšího, Tailwind poskytuje vývojářům maximální flexibilitu při definování vzhledu a chování jejich stránek.
5. Optimalizace Velikosti: Díky použití pouze těch utility tříd, které jsou skutečně potřebné, může Tailwind vytvářet optimálně velké CSS soubory, což přispívá k rychlejšímu načítání stránek.

Vývojáři, kteří upřednostňují rychlost a jednoduchost při stylování svých projektů, ocení Tailwind CSS jako mocný nástroj pro efektivní práci s CSS.

2.1.4 Navigační panel (Navbar)

Navigační panel byl jako většina tohoto internetového obchodu vytvořen dynamicky, to znamená že jsou data jako odkazy na stránky, texty, které se vypíší na bar a informace o tom, zda se daná stránka má otevřít v samostatném okně uložena v databázi odkud se získávají pomocí prisma a funkce map. Následně se odesílají do komponenty „Navbar“, kde vkládají se vykreslují na stránku pomocí props.



Obr. 5: Navigační lišta

```
const links = await prisma.navbarLinks.findMany()

return (
  <div className='flex h-fit w-full justify-between border-b ■border-gray-900 □bg-white top-0 sticky z-50 '>
    <div className='flex w-40 relative '>
      <Link
        href="/"
      >
      <Image
        src="/logo.png"
        alt="logo.png"
        fill
        className='object-contain py-1'
      />
    </Link>
  </div>

  <div className='flex'>
    <div className='flex items-center'>
      {
        links.map((item, index) => {
          return (
            <Link
              href={item.link}
              key={index}
              className='leading-normal py-6 px-5'
              target={item.external ? '_blank': '_self'}
            >
              {item.text}
            </Link>
          )
        })
      }
    </div>

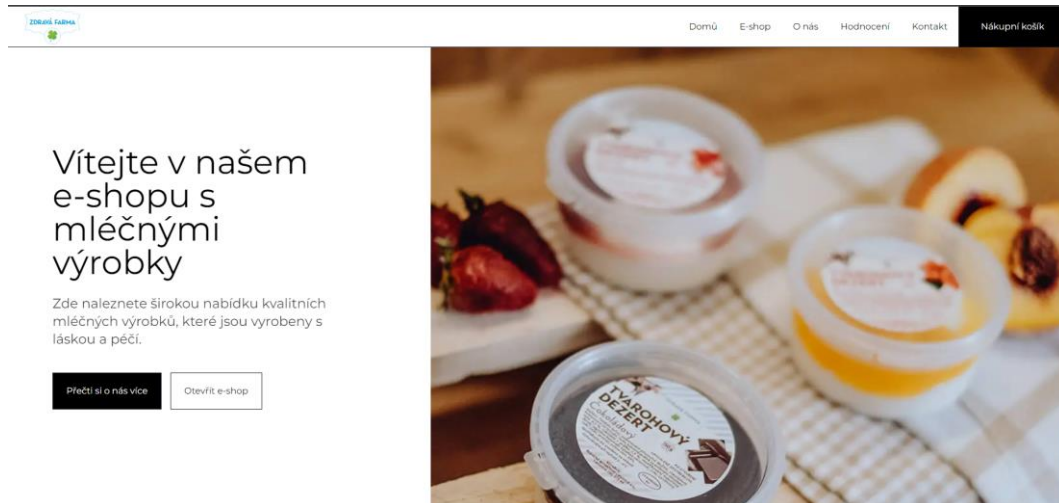
    <div>
      <Link
        href={"/kosik"}
        className='flex py-6 px-10 ■bg-black items-center □text-white'
      >
        Nákupní košík
      </Link>
    </div>
  </div>
)

```

Obr. 6: Kód navigační lišty

2.1.5 Úvodní stránka

Na úvodní stránku byl zvolen jednoduchý design a rozložení stránky. Potenciální zákazníci přivítá uvítací titulek s informacemi, co na zdejším e-shopu naleznou. Dominantou stránky je automatická prezentace náhodných produktů z nabídky obchodu.



Obr. 7: Úvodní stránka

Prezentace výrobků se provádí dvěma komponentami. Na úvodní straně se zavolá komponenta „ImageSliderServer“ ve které, se z databáze vezmou URL adresy obrázků které se zde promítají. Následně se zavolá samotná komponenta „ImageSlider“ do které, se odešle pole Stringů s danými adresami. Zde se pole projde pomocí map() a pomocí scriptu se dané obrázky procházejí.

```
const ImageSlider = (props: Props) => {
  const [currentImage, setCurrentImage] = useState(0);

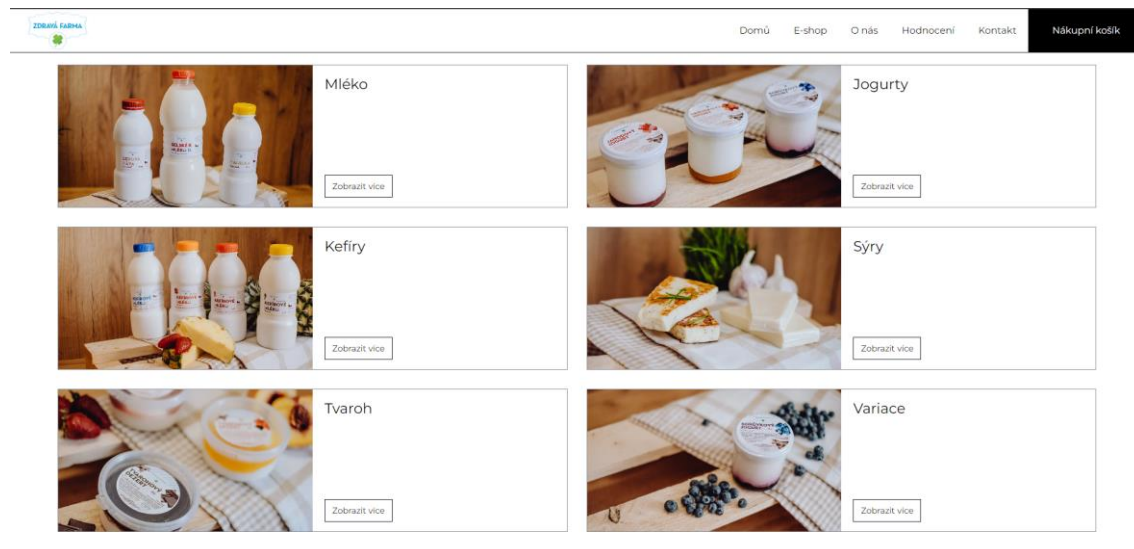
  useEffect(() => {
    const interval = setInterval(() => {
      setCurrentImage((prevImage) => (prevImage + 1) % props.images.length);
    }, 3000);
    return () => clearInterval(interval);
  }, []);

  return (
    <div className="flex relative w-full h-full">
      {
        props.images.map((image, index) => {
          if(currentImage == index){
            return(
              <Image
                src={image.image}
                key={index}
                alt=""
                className="object-cover object-top"
                fill
              />
            )
          }
        })
      }
    </div>
  );
};
```

Obr. 8: Prezentace obrázků

2.1.6 Produkty

Stránka s kategoriemi je také dynamicky generovaná. V hlavním souboru page.tsx se z databáze vytáhnou všechna dostupná data o kategoriích zboží. Pomocí funkce map se projde proměnná categories, ve které se nachází vybraná data a postupně se zavolá komponenta „CategoryCard“. Ta dle příchozích parametrů postupně vypíše všechny kategorie. Součástí každé této karty se je obrázek dané kategorie v jednom „divu“ a název, popis a proklikávací tlačítko ve druhém „divu“.

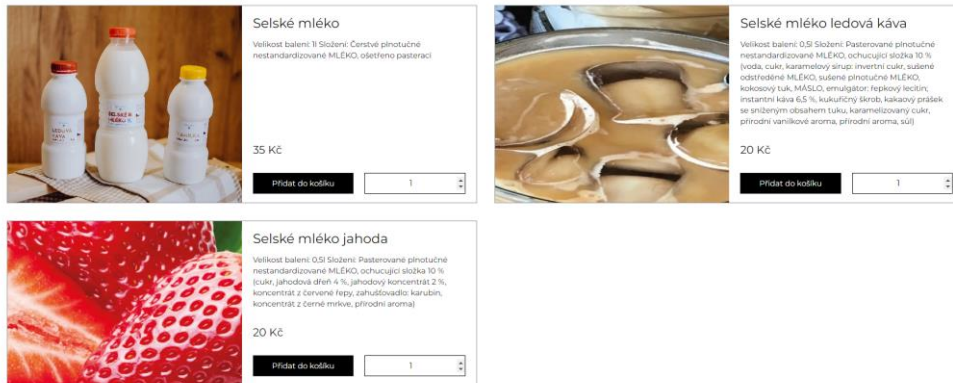


Obr. 9: Stránka s kategoriemi

```
5
6 type Props = {
7   image: string;
8   title: string;
9   description: string;
10  slug: string;
11 }
12
13 export default function CategoryCard(props: Props) {
14   return (
15     <div className='flex w-full min-h-60 border border-gray-900 border-opacity-50'>
16       <div className='flex w-1/2 h-full relative'>
17         <Image
18           src={props.image}
19           alt={props.title}
20           fill
21           className='object-cover'
22         />
23       </div>
24
25       <div className='flex flex-1 flex-col gap-3 px-5 py-4'>
26         <p className='text-2xl'>{props.title}</p>
27         <p className='text-sm'>{props.description}</p>
28         <div className='mt-auto pt-4'>
29           <Button href={` /kategorie/${props.slug}`} size="small">Zobrazit více</Button>
30         </div>
31       </div>
32     </div>
33   )
34 }
```

Obr. 10: Komponenta categoryCard

Kliknutím na jednotlivou kategorii, se uživatel dostane na kategorii, kterou si vybral. Všechny kategorie jsou generovány dynamicky. V podsložce „[kategorie_slug]“ se nachází template pro výpis produktů v kategorii. Kategorie, která se má vypsat, se určuje dle „slugu“, který se nachází v tlačítku u dané kategorie. Ten se následně použije jako parametr pro vyhledávání v databázi. Odtud se vyberou potřebná data a ty se postupně vloží do komponenty „ProductCard“.



Obr. 11: Stránka s produkty

```

return (
  <div className='flex w-full min-h-60 border border-gray-900 border-opacity-50'>
    <div className='flex w-1/2 h-full relative'>
      <Image
        src={props.image}
        alt={props.title}
        fill
        className='object-cover'
      />
    </div>

    <div className='flex flex-1 flex-col gap-3 px-5 py-4'>
      <p className='text-2xl'>{props.title}</p>
      <Markdown className='text-sm'>
        {props.description}
      </Markdown>
      {
        props.cena == 0 ? (
          <p className='text-xl mt-auto pt-4'>Zdarma</p>
        ) : (
          <p className='text-xl mt-auto pt-4'>{props.cena} Kč</p>
        )
      }
      <div className='pt-4'>
        <AddToCart
          key={props.id}
          id={props.id}
          pocet={props.pocet}
          cena={props.cena}
        />
      </div>
    </div>
  </div>
)

```

Obr. 12: Kód s výpisem produktů

2.1.7 ADW

Stránka nazvaná ADW se zabývá jak obecnými informacemi, tak následně výčtem z historie této firmy. Jedna z podstránek této sekce je taktéž galerie s fotografiemi z normálního dne této rozvinuté firmy. Druhou záložkou jsou podrobnější informace. Nachází se zde příklady postupů při od pěstování rostlin, přes živočišnou výrobu až po samotné zpracování mléka pro výrobu daných produktů, jenž se prodávají na tomto internetovém obchodu.

Vítejte u ADW - Naše Farma

Zabýváme se tradiční zemědělskou prvovýrobou. Pěstujeme tržní plodiny a věnujeme se chovu skotu, zejména mléčného. Hospodáříme pod hlávkou ROLNICKÉ SPOLEČNOSTI LESONICE na Vysočině na Třebíčsku, ADW FARM v Jihomoravském kraji na Ivančicku a Zbraslavsku a v okolí Dalešic na farmě AGRO MOHELNO.

Více o nás

Navštívit galerii



Film a TV pořady

Obr. 13: Náhled stránky s popisem ADW

Galerie fotek ADW



Minerální hnojení



Tatra při vápnění



Obr. 14: Galerie

```

type Props = {
  image: string;
  title: string;
}

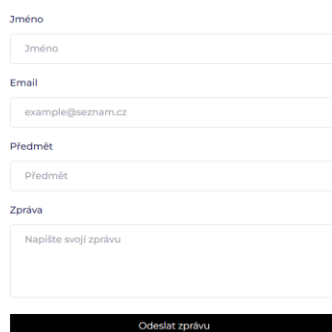
export default function GalleryCard(props: Props) {
  return (
    <div className='flex flex-col group justify-center pt-7'>
      <div className='h-96 relative'>
        <Image
          src={props.image}
          alt='food'
          fill
          className='object-cover'
        />
      </div>
      <div>
        <h3 className='text-2xl font-semibold mt-4'>
          {props.title}
        </h3>
      </div>
    </div>
  )
}

```

Obr. 15: Komponenta GalleryCard

2.1.8 Kontakt

Jako poslední položka na navigační liště mimo košíku, se nachází stránka s kontaktem. Všechny důležité kontakty na vedoucí hlavní pobočky, která se nachází v obci Lesonice se nachází v levé části patičky. Hlavní částí této stránky je kontaktní formulář, který odešle potřebnou zprávu na e-mailovou adresu hlavní pobočky.



The image shows a contact form with the following fields and a button:

- Jméno**: Input field containing "Jméno".
- Email**: Input field containing "example@seznam.cz".
- Předmět**: Input field containing "Předmět".
- Zpráva**: Textarea containing "Napíšte svojí zprávu".
- Odeslat zprávu**: A black button with white text.

Obr. 16: Kontaktní stránka

Kontaktní formulář odešle vyplněná data na API pomocí funkce „fetch“, zde se daná data zpracují a odešlou pomocí internetové služby Resend. Po kliknutí na tlačítko „Odeslat zprávu“ se hodnoty z formuláře uloží do useState a odtud se funkci fetch odešlou přes URL adresu do API. Zde se data pro přehlednost mailů uloží do databáze do tabulky „Emails“ a následně se odešlou přes Resend na přednastavený e-mail.

```
11
12  const resend = new Resend(process.env.RESEND_API_KEY);
13
14  export async function GET(
15    req : NextRequest,
16    { params } : Params
17  ) {
18
19    const jmeno = req.nextUrl.searchParams.get("jmeno")
20    const hlavicka = req.nextUrl.searchParams.get("hlavicka")
21    const email = req.nextUrl.searchParams.get("email")
22    const message = req.nextUrl.searchParams.get("message")
23
24    const data = await prisma.emails.create({
25      data: {
26        jmeno: String(jmeno),
27        prijmeni: String(hlavicka),
28        email: String(email),
29        zprava: String(message)
30      }
31    })
32
33    try {
34      const data = await resend.emails.send({
35        from: 'Acme <onboarding@resend.dev>',
36        to: ['venhoda@student.vspj.cz'],
37        subject: String(hlavicka),
38        text: String(message)
39      });
40
41      return Response.json(data);
42    } catch (error) {
43      return Response.json({ error });
44    }
45  }
46
```

Obr. 17: API pro odeslání e-mailu

2.1.9 Košík

Košík je vytvořen pomocí Cookies. Při prvním přidání zboží do košíku, se Id daného produktu odešle do API.

```
fetch(`/api/product/${props.id}?pocet=${value}`);
```

Obr. 15: Odeslání dat do API

Následně se zkontrolují cookies, jestli již existuje „kosikId“ což je Id daného košíku. Na ID košíku jsem použil UUID. Pokud daný košík neexistuje, vytvoří se nový záznam v databázi v tabulce „objednávky“. Tento záznam slouží k propojení daného košíku se zbožím v něm. V dalším kroku se do tabulky „objednávkyItems“ uloží produkt, jehož ID a počet se přenesli v adrese.

```
export default function AddToCart(props: Props) {  
  const [value, setValue] = useState("1")  
  
  const onClick = () => {  
    if(Number(value) == 0)  
    {  
      alert("zadejte počet kusů");  
    }  
    else{  
      fetch(`/api/product/${props.id}?pocet=${value}&cena=${props.cena}`);  
    }  
  }  
  
  if(props.pocet != 0){  
    return (  
      <div className='grid grid-cols-2 gap-5 justify-between'  
        <div className='flex border border-gray-900 bg-black text-white items-center justify-center w-full px-3 py-2 text-sm cursor-pointer' onClick={()=> {onClick()}}>  
          Přidat do košíku  
        </div>  
        <input className='flex border border-black w-full text-center '  
          value={value} type="number" id="pocet" name="pocet" defaultValue={1} min={0} max={props.pocet}  
          onChange={(e) => setValue(e.target.value)}  
        />  
      </div>  
    )  
  }  
}
```

Obr. 18: Vytvoření košíku a přidání zboží

Po kliknutí na tlačítko košík v pravém horním rohu se na stránku vypíše obsah momentálního košíku. V případě že zákazník dosud nevlozil žádné zboží, vypíše se nápis „Košík je prázdný“.



Obr. 19: Výpis zboží v košíku

Zde se z databáze získají data, která obsahují Id momentálního košíku. Položku po položce se odešlou do komponenty „cartCard“, která je vykreslí na stránku. V této komponentě se zároveň vykreslí i tlačítka „Smazat“ a input typu number kterým, jde upravovat požadovaný počet produktů. Tyto tlačítka odesílají ID daného záznamu do API, kde se záznamy v databázi aktualizují použitím funkcí Update a Delete v Prismě.

```

5  ) {
6
7  console.log(params.product_id);
8
9  const cookie = cookies();
0  const kosikId= cookie.get('kosikId');
1  let kosik_id = kosikId?.value;
2
3  const inKosik = await prisma.objednavkyItems.findFirst({
4    where:{
5      productId: Number(params.product_id),
6      objednavkyId: kosik_id
7    },
8    select:{
9      id:true
0    }
1  })
2
3  const data = await prisma.objednavkyItems.delete({
4    where:{
5      id: inKosik?.id,
6    }
7  })
8
9  return NextResponse.json({
0    message: "Hello World! Delete"
1  })
2  }
3

```

Obr. 20: API na smazání produktu v košíku.

Tlačítkem na konci stránky se uživatel posune dále v objednávce. Dostane se na stránku, kde vyplní všechny potřebné údaje pro doručení objednávky jako je například jméno, e-mail, ulice atd. Tyto údaje se dále odešlou do API na zpracování a odeslání e-mailu s potvrzením objednávky. V API souboru se zpracují data z formuláře a uloží se do proměnné „message“. Dále se zde podle UUID v cookies vytáhnou všechny produkty, které byly vloženy do košíku a pomocí funkce „map“ se uloží do proměnné „items“ s vypočítanou cenou každého produktu a s počtem objednaných produktů. Tyto 2 proměnné se následně spojí a odešlou se přes službu Resend jak na e-mail zákazníka, tak na e-mail společnosti. Navíc se daná objednávka uloží do databáze do tabulky „objednavkyHistorie“ a upraví se stávající počet zboží v tabulce „produkty“.

```

const kosik = await prisma.objednavkyItems.findMany({
  where: {
    objednavkyId: cookie,
  },
  select: {
    productId: true,
    product: true,
    pocet: true,
    id: true,
  }
})

let items = "";
let finalCena = 0;
{
  kosik.map((item, index) => {
    finalCena = item.pocet * item.product.price;
    items += String(item.product.name) + " " + String(item.pocet) + " ks " + finalCena + " Kč \n"
  })
}

const message = jmeno + "\n" + email + "\n" + ulice + "\n" + mesto + "\n" + psc + "\n" + telefon + "\nVaše objednávka: \n" + items

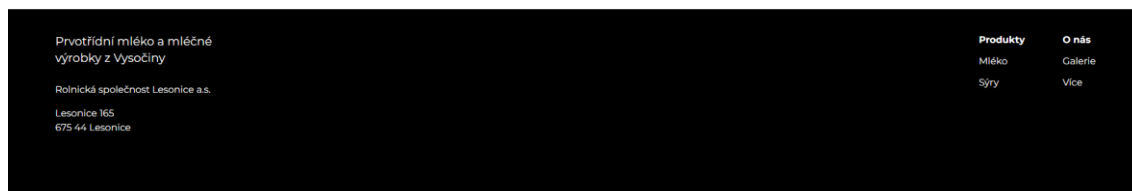
try {
  const data = await resend.emails.send({
    from: 'Acme <onboarding@resend.dev>',
    to: ['venhoda@student.vspj.cz'],
    subject: "Váš nákup z obchodu zdravafarma",
    text: String(message)
  });
}

```

Obr. 21: odeslání objednávky

2.1.10 Patička

V patičce na spodní straně stránky nalezneme zjednodušené kontaktní informace, odkazy na sociální sítě a jednoduchou navigační lištu na přesnější pohyb po stránce, například přímo na část ADW galerie. Její pravá část s navigací je generovaná dynamicky. Údaje pro generování odkazů jsou uloženy v poli mimo funkci a následně vykreslovány pomocí map do samotné patičky.



Obr. 22: Patička

```

5
7 export default function Footer() {
3   return (
3     <div className='flex w-full justify-between px-20 min-h-80 pt-10 ■bg-black □text-white'>
3       <div className='flex flex-col max-w-72 gap-4'>
1         <p className='text-xl'>Prvotřídní mléko a mléčné výrobky z Vysočiny</p>
2         <p className='pt-3'>Rolnická společnost Lesonice a.s.</p>
3         <p className=''>Lesonice 165 <br/> 675 44 Lesonice</p>
1       </div>
5
5       <div className='flex gap-16'>
7         {
3           columns.map((column, index) => {
3             return(
3               <div className='flex flex-col gap-3'>
1                 <p className='font-bold'>{column.title}</p>
2                 {
3                   column.links.map((link, index) => {
1                     return(
5                       <Link href={link.href} className='' target={link.external ? '_blank': '_self'}>{link.text}</Link>
5                     )
7                   })
3                 }
3               </div>
1             ))
2           })
3         }
1       </div>
5     </div>
5   )
7 }

```

Obr. 23: Generování patičky

2.2 Implementace back-endu

Implementace back-endové části e-commerce platformy představuje klíčový aspekt vytváření efektivního, bezpečného a spolehlivého systému pro zpracování dat, řízení logiky aplikace a komunikaci s databází. Následující kapitola podrobně popisuje technologický stack| a implementační postup použitý při vývoji back-endové části naší e-commerce platformy.

2.2.1 MySQL databáze

Databázový systém e-shopu je integrován pomocí databázového systému Prisma. Systém se do projektu implementoval přes instalační balíčky npm. Rozkládá se ve třech samostatných souborech a to v souborech prisma.ts, schema.prisma a soubor .env. V souboru prisma.ts se nachází PrismaClient, což je „aplikační rozhraní“ a je definováno na základě děfinice datových modelů. Generuje TypeScript typy na základě struktury databáze, což poskytuje vývojářům silnou typovou kontrolu přímo v rámci kódu. Soubor schema.prisma, obsahuje modelovou definici databáze systému. Každý model reprezentuje jednu tabulku v databázi. Jednotlivé modely obsahují atributy, které odpovídají sloupcům v příslušných tabulkách.

```

model User {
  id      Int      @id          @default(autoincrement())
  email   String   @unique       @db.VarChar(255)
  name    String   @db.VarChar(255)
  surname String   @db.VarChar(255)
  pass    String   @db.VarChar(255)
}

model Categories {
  id      Int      @id          @default(autoincrement())
  name    String   @db.VarChar(255)
  slug    String   @db.VarChar(255)
  image   String   @db.VarChar(255)
  description String @db.Text()

  products Products[]
}

```

Obr. 24: Modely databáze

2.2.2 UploadThing

Všechny obrázky používané v tomto projektu jsou uloženy na stránce UploadThing. Odtud se následně přivolávají do systému a vykreslují na webu. Nejprve jsem musel nainstalovat potřebné balíčky.

```

npm install uploadthing @uploadthing/react

```

Obr. 25: Instalace uploadthing knihovny

Jako další jsem musel vytvořit API soubory „route.ts“ a „core.ts“. Tyto soubory zajišťují připojení na servery UploadThing a nahrání obrázku. URL adresa obrázku se následně vypíše do console VS code.

```
app > api > uploadthing > TS core.ts > ...
 1 import { createUploadthing, type FileRouter } from "uploadthing/next";
 2 import { UploadThingError } from "uploadthing/server";
 3
 4 const f = createUploadthing();
 5
 6 const auth = (req: Request) => ({ id: "fakeId" }); // Fake auth function
 7
 8 // FileRouter for your app, can contain multiple FileRoutes
 9 export const ourFileRouter = {
10   // Define as many FileRoutes as you like, each with a unique routeSlug
11   imageUploader: f({ image: { maxFileSize: "16MB" } })
12     // Set permissions and file types for this FileRoute
13     .middleware(async ({ req }) => {
14       // This code runs on your server before upload
15       const user = await auth(req);
16
17       // If you throw, the user will not be able to upload
18       if (!user) throw new UploadThingError("Unauthorized");
19
20       // Whatever is returned here is accessible in onUploadComplete as `metadata`
21       return { userId: user.id };
22     })
23     .onUploadComplete(async ({ metadata, file }) => {
24       // This code RUNS ON YOUR SERVER after upload
25       console.log("Upload complete for userId:", metadata.userId);
26
27       console.log("file url", file.url);
28
29       // !!! Whatever is returned here is sent to the clientside `onClientUploadComplete` callback
30       return { uploadedBy: metadata.userId };
31     }),
32 } satisfies FileRouter;
33
34 export type OurFileRouter = typeof ourFileRouter;
```

Obr. 26: Soubor core.ts

Dále jsem musel vytvořit komponentu pro „UploadButton“, která po kliknutí vytvoří okno, ve kterém najdete potřebný obrázek a následně ho nahraje na stránku.

2.2.3 AdminJS

Pro management systému ze strany zaměstnanců obchodu jsem využil službu AdminJs. S pomocí této služby uživatel získá jednoduchý systém na správu databáze. Prvním krokem byla instalace knihovny.

```
$ npm i -g @adminjs/cli
```

Obr. 27: Instalace AdminJS

Dalším krokem bylo nastavení adaptéru Prisma v souboru index.ts a nastavení údajů o databázi v souboru „.env“. Dále jsem musel vzít nastavení samotné databáze příkazem „npx prisma db pull“ a vytvoření prisma klienta přes „npx prisma generate“. Jako poslední před spuštěním admina bylo potřeba nastavit tabulky, které jsem potřeboval v adminovi ovládat. To jsem udělal v souboru „options.ts“ ve funkci „AdminJSOptions“ v parametru „resources“.

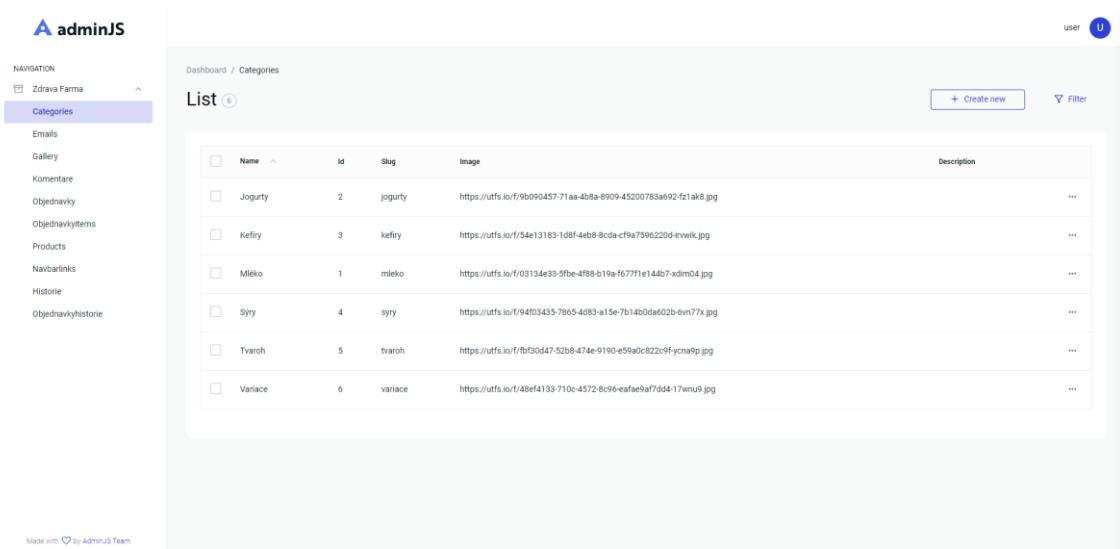
```

.8 }
.9
!0 const options: AdminJSOptions = {
!1   componentLoader,
!2   rootPath: '/admin',
!3   resources: [
!4     {
!5       resource: { model: getModelByName('categories'), client: prisma },
!6       options: {
!7         parent: parent,
!8       },
!9     },
!0   ],

```

Obr. 28: Nastavení zdrojů pro AdminJS

Nakonec se admin musel buildnout do aktivních souborů příkazem „npm run build“.



Obr. 29: AdminJS

Závěr

V rámci bakalářské práce byl úspěšně vytvořen e-shop pro zemědělské družstvo pomocí technologií React.js, Node.js a Next.js. Tento projekt ukázal, jak tyto moderní technologie mohou být efektivně využity k vývoji komplexních webových aplikací.

React.js, jako základní stavební kámen tohoto projektu, umožnil vytvoření uživatelského rozhraní, které je nejen intuitivní a responzivní, ale také snadno udržitelné díky jeho komponentovému designu. Díky stavovému managementu pomocí kontextu a háků bylo možné efektivně sledovat a manipulovat s daty napříč komponentami, což je zásadní pro funkčnost e-shopu.

Next.js, postavený na React.js a Node.js, přinesl do projektu podporu pro server-side rendering a statické generování stránek. Tyto funkce nejen zlepšily výkon aplikace, ale také optimalizovaly její viditelnost ve vyhledávačích, což je klíčové pro úspěch jakéhokoli e-shopu. Díky routování na úrovni souborů a podpoře pro API routy bylo možné snadno navigovat a spravovat aplikaci a efektivně manipulovat s daty.

Node.js, jako backendová platforma, umožnil vytvoření robustního a bezpečného serveru pro správu dat e-shopu. Jeho asynchronní, event-driven architektura zajišťovala vysokou propustnost a efektivitu, což je nezbytné pro správu velkého množství uživatelských požadavků a transakcí.

Celkově vzato, tento projekt ukázal, jak kombinace React.js, Node.js a Next.js může poskytnout silný základ pro vývoj moderních webových aplikací. E-shop pro zemědělské družstvo, který byl vytvořen v rámci práce, nejen splňuje všechny požadavky a očekávání, ale také demonstruje, jak tyto technologie mohou být využity k vytvoření řešení, které je škálovatelné, udržitelné a připravené na budoucí rozšíření a vylepšení.

V budoucnu by mohlo být užitečné prozkoumat další možnosti a nástroje, které tyto technologie nabízejí, jako je například inkrementální statické generování v Next.js nebo serverless architektura v Node.js. Také by bylo zajímavé prozkoumat integraci s dalšími technologiemi a platformami, jako jsou databáze, cloudové služby nebo umělá inteligence, aby se ještě více rozšířila funkcionální a efektivita e-shopu.

V závěru lze říci, že tento projekt byl úspěšným důkazem konceptu a ukázal, jak mohou být moderní webové technologie efektivně využity k řešení reálných problémů a potřeb. Je to inspirace pro další vývojáře a ukazuje cestu k budoucnosti webového vývoje.

Seznam použité literatury

- [Co je] Vývojář Full Stack + 9 kroků, jak se stát jedním (2024). Online. 2022. Dostupné z: <https://www.collectiveray.com/cs/full-stack-developer>. [cit. 2024-01-03].
- 5 e-commerce trendů, které provázejí rok 2023 [online]. 2023 [cit. 2024-01-03]. Dostupné z: <https://www.lemonero.cz/blog/trendy-ecommerce-2023>
- ADW. Online. 2024. Dostupné z: <https://www.adw.cz/>. [cit. 2024-03-30].
- Definice a Kodex farmářských trhů - Vitalia.cz. [online] Available at: <https://www.vitalia.cz/texty/definice-a-kodex-farmarskych-trhu/> [Accessed 24 Nov. 2023]
- MySQL [online]. 2024 [cit. 2024-03-29]. Dostupné z: <https://www.mysql.com/>
- Next.js by Vercel. Online. 2024. Dostupné z: <https://nextjs.org/>. [cit. 2024-03-29].
- Node.js Online. 2024. Dostupné z: <https://nodejs.org/en>. [cit. 2024-01-03].
- Prisma. Online. 2024. Dostupné z: <https://www.prisma.io/>. [cit. 2024-03-29].
- PROCHÁZKOVÁ, Alena. PŘÍNOS FARMÁŘSKÝCH TRHŮ PRO ČESKÉ FARMÁŘE. Bakalářská práce. Praha: Univerzita Karlova, 2012.
- React + Node.js + Express + MySQL example: Build a CRUD App. Online. 2023. Dostupné z: <https://www.bezkoder.com/react-node-express-mysql/>. [cit. 2024-01-03].
- React. Online. 2024. Dostupné z: <https://react.dev/>. [cit. 2024-01-03].
- Resend. Online. 2024. Dostupné z: <https://resend.com/>. [cit. 2024-03-29].
- Trendy a vývoj e-commerce pro zemědělské produkty. [Online]. Available: <https://www.eagri.cz/public/portal/mze/ministerstvo-zemedelstvi/proexportni-okenko/aktualni-trendy-v-oblasti-produktu>. [Accessed: 24-Nov-2023].
- Trendy v grafickém designu roku 2023: Brutalismus. Online. Desing portal. 2023. Dostupné z: <https://www.designportal.cz/trendy-v-grafickem-designu-roku-2023-brutalismus/>. [cit. 2024-03-28].
- Uploadthing. Online. 2024. Dostupné z: <https://uploadthing.com/>. [cit. 2024-03-29].
- W3Schools. Online. 2024. Dostupné z: <https://www.w3schools.com/>. [cit. 2024-01-03].
- Zdravá Farma. Online. 2024. Dostupné z: <https://www.zdravafarma.cz/>. [cit. 2024-03-30].