

Vývoj informačního systému pro evidenci kulturních akcí

Josef Bouchal

Bakalářská práce
2024



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Josef Bouchal**
Osobní číslo: **A21320**
Studijní program: **B0613A140020 Softwarové inženýrství**
Forma studia: **Prezenční**
Téma práce: **Vývoj informačního systému pro evidenci kulturních akcí**
Téma práce anglicky: **Development of an Information System for Recording Cultural Events**

Zásady pro vypracování

- Popište současný stav dané problematiky.
- Popište technologie, které budou v práci použité.
- Provedte analýzu požadavků pro evidenci kulturních akcí.
- Navrhněte a realizujte databázi a aplikaci.
- Provedte otestování aplikace.
- Zhodnoťte provedené řešení a navrhněte případné vylepšení.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. DEAN, John, c2019. *Web programming with HTML5, CSS, and JavaScript*. Burlington: Jones & Bartlett Learning. ISBN 9781284091793.
2. PRICE, Mark J., 2022. *C# 11 and .NET 7 – Modern Cross-Platform Development Fundamentals: Start Building Websites and Services with ASP.NET Core 7, Blazor, and EF Core 7*. 7th edition. Birmingham: Packt Publishing. ISBN 978-1-80323-780-0.
3. SHIELDS, Walter, c2016. *SQL QuickStart Guide: The Simplified Beginner's Guide to Managing, Analyzing, and Manipulating Data With SQL*. Albany: ClydeBank Media. ISBN 9781945051753.
4. *Dokumentace k jazyku C#*, 2023. Online. Dokumentace pro .NET | Microsoft Learn. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/csharp/>. [cit. 2023-11-09].
5. *Introduction*, 2023. Online. Bootstrap. Dostupné z: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>. [cit. 2023-11-09].

Vedoucí bakalářské práce:

Ing. Dušan Fojtů, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

5. listopadu 2023

Termín odevzdání bakalářské práce:

13. května 2024

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 5. ledna 2024

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.
- že při tvorbě této práce jsem použil nástroj generativního modelu AI, konkrétně ChatGPT, dostupný na webové adrese <https://www.openai.com/chatgpt> za účelem naplnění databáze daty pro demonstraci webové stránky Cultur4U. Po použití tohoto nástroje jsem provedl kontrolu obsahu a přebírám za něj plnou zodpovědnost.

Ve Zlíně, dne 13.5.2024

Josef Bouchal, v.r.
podpis studenta

ABSTRAKT

Bakalářská práce se zabývá vývojem informačního systému pro evidenci kulturních akcí. Teoretická část popisuje aktuální stav v dané problematice a dostupná řešení na trhu. Dále představuje technologie a knihovny, které byly v práci použity. Praktická část popisuje vývoj webové aplikace, zahrnující analýzu požadavků, její vytvoření na platformě ASP.NET Core MVC včetně implementace databáze a následnou publikaci pomocí Azure. Systém umožňuje organizátorům správu akcí a generování závěrečných zpráv, uživatelé mohou prohlížet a vyhledávat akce, včetně možnosti zobrazení kalendáře akcí. Na závěr práce jsou zhodnoceny dosažené výsledky a navrženy možnosti budoucího rozvoje aplikace.

Klíčová slova:

ASP.NET Core, SQL, C#, Azure, webová aplikace, evidence kulturních akcí, informační systém, Bootstrap

ABSTRACT

This bachelor's thesis deals with the development of an information system for the registration of cultural events. The theoretical part describes the current state of the field and the solutions available on the market. It also presents the technologies and libraries that were used while developing the application. The practical part describes the development of a web application, including requirements analysis, its creation on the ASP.NET Core MVC platform, including database implementation, and subsequent publishing using Azure. The system allows organizers to manage events and generate final reports, and users to view and search for events, including the ability to view the calendar of events. The thesis concludes with an evaluation of the results achieved and proposes possibilities for the future development of the application.

Keywords:

ASP.NET Core, SQL, C#, Azure, web application, cultural events registration, information system, Bootstrap

Chtěl bych poděkovat vedoucímu mé bakalářské práce panu Ing. Dušanovi Fojtů, Ph.D., za jeho rady, trpělivost a čas, který mi v průběhu zpracování bakalářské práce poskytl. Dále bych chtěl poděkovat své rodině, přítelkyni a přátelům za podporu.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 SOUČASNÝ STAV PROBLEMATIKY	12
1.1 FACEBOOK EVENTS	13
1.1.1 Výhody	14
1.1.2 Nevýhody	14
1.2 KUDY Z NUDY	15
1.2.1 Výhody	15
1.2.2 Nevýhody	15
1.3 PRAGOKONCERT BOHEMIA, A. S.	16
1.3.1 Výhody	16
1.3.2 Nevýhody	16
2 POUŽITÉ NÁSTROJE A TECHNOLOGIE	17
2.1 VÝVOJOVÉ PROSTŘEDÍ	17
2.1.1 Visual Studio 2022	18
2.2 DATABÁZE	18
2.2.1 Relační databáze (SQL)	19
2.2.2 Nerelační databáze (NoSQL)	20
2.2.3 Hierarchické databáze	20
2.2.4 Síťové databáze	20
2.2.5 Objektově orientované databáze	20
2.3 SQL SERVER MANAGEMENT STUDIO (SSMS)	20
2.4 HTML.....	21
2.5 CSS.....	22
2.6 JAVASCRIPT	23
2.7 ASP.NET CORE	23
2.7.1 C#	23
2.7.2 Blazor	24
2.7.3 Entity Framework Core	25
2.7.4 Identity v ASP.NET Core	26
2.7.5 Návrhový vzor MVC.....	27
2.8 BOOTSTRAP	28
2.9 AZURE.....	30
2.9.1 Azure App Service	31
2.9.2 Azure SQL Database	31
2.9.3 Úložiště Azure (Azure Storage)	31
3 POUŽITÉ KNIHOVNY	32
3.1 QUILLJS	32
3.2 FULLCALENDAR	32
3.3 CHART.JS.....	32
3.4 ITEXT7	33
4 TESTOVÁNÍ	34

4.1	DRUHY TESTOVÁNÍ PODLE ZNALOSTI KÓDU	34
4.1.1	White box testování.....	34
4.1.2	Black box testování	34
4.1.3	Gray box testování	34
4.2	DRUHY TESTOVÁNÍ PODLE ÚROVNĚ	35
4.2.1	Unit testy (jednotkové testování)	35
4.2.2	Integrační testování	35
4.2.3	Systémové testování.....	35
4.2.4	Akceptační testování	35
II	PRAKTICKÁ ČÁST	36
5	ANALÝZA POŽADAVKŮ	37
5.1	FUNKČNÍ POŽADAVKY	37
5.2	NEFUNKČNÍ POŽADAVKY	38
6	NÁVRH A DATABÁZE	39
6.1	TABULKA ASPNETUSERS.....	39
6.2	TABULKA CULTURUEVENTS	40
6.3	TABULKA EVENTUSERS	40
6.4	TABULKA TICKETS	40
6.5	TABULKA USERCOMPANIES	40
6.6	TABULKA ATTACHMENTS	40
6.7	TABULKA COMPANIES.....	40
6.8	IMPLEMENTACE DATABÁZE	41
7	WEBOVÁ APLIKACE CULTUR4U	43
7.1	NAVIGACE.....	43
7.2	DOMOVSKÁ STRÁNKA	45
7.3	PŘIHLÁŠENÍ.....	47
7.4	REGISTRACE.....	47
7.4.1	Odesílání verifikačního kódu	48
7.5	DETAIL AKCE	50
7.5.1	Tvorba závěrečné zprávy	50
7.6	NASTAVENÍ SPOLEČNOSTI	52
7.7	NASTAVENÍ UŽIVATELE.....	56
7.8	SEZNAM AKCÍ.....	56
7.8.1	Formulář pro přidání akce.....	57
7.8.1.1	Ukládání souborů v akci	59
7.8.2	Formulář pro editaci akce.....	60
7.8.3	Statistiky akce	61
7.9	KALENDÁŘ AKCÍ	63
8	TESTOVÁNÍ V APLIKACI CULTUR4U.....	65
9	PUBLIKACE APLIKACE DO MICROSOFT AZURE	68
10	ZHODNOCENÍ A MOŽNÁ VYLEPŠENÍ.....	70

10.1	ZHODNOCENÍ VÝSLEDKŮ	70
10.2	MOŽNOSTI VYLEPŠENÍ APLIKACE	70
10.2.1	Lepší filtrování akcí	70
10.2.2	Rozšíření statistik	71
10.2.3	Prodej vstupenek	71
10.2.4	Rozšíření detailu akce	71
10.2.5	Sdílení akcí na sociální síť	71
10.2.6	Rozšíření tvorby závěrečné zprávy	71
	ZÁVĚR	72
	SEZNAM POUŽITÉ LITERATURY	73
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	77
	SEZNAM OBRÁZKŮ	78
	SEZNAM PŘÍLOH.....	80

ÚVOD

Ve světě, kde se digitalizace stává neodmyslitelnou součástí každodenního života, představují kulturní akce klíčový prostředek k spojování lidí, podpoře společenské interakce a sdílení kulturních hodnot. Historicky byly informace o kulturních akcích šířeny prostřednictvím tištěných materiálů, jako jsou plakáty a letáky, což bylo časově náročné a méně efektivní. V dnešní době, kdy technologie pronikly do všech sfér našeho života, přichází nová éra propagace a správy kulturních akcí prostřednictvím informačních systémů.

Pandemie COVID-19 měla značný dopad na kulturní sektor po celém světě. Vládní opatření, která byla zavedena s cílem zamezit šíření viru, vedla k rozsáhlému omezení nebo úplnému zrušení veřejných kulturních akcí. Tato situace dramaticky ukázala, jak důležité jsou kulturní akce pro společnost. Lidé se ocitli izolovaní ve svých domovech, a právě kulturní akce, ať už online nebo v omezeném formátu, jim poskytovaly pocit společenství.

Pro organizátory jsou informační systémy nepostradatelným nástrojem, který umožňuje získávání dat o úspěšnosti akce, zjednodušuje propagaci a celkový marketing akce. Mohou nabízet rovněž funkce pro generování závěrečných zpráv, které jsou klíčové pro vyúčtování a získávání finančních prostředků z různých kulturních fondů. Z pohledu účastníků kulturních událostí informační systémy zjednodušují vyhledávání akcí a získávání dalších informací.

Tato bakalářská práce se zaměřuje na vývoj a implementaci informačního systému pro evidenci kulturních akcí, který reaguje na současné potřeby kulturního sektoru a zároveň reflektuje technologický pokrok a změny ve společenských trendech. Cílem práce je vytvoření informačního systému pro evidenci kulturních akcí, který organizátorům umožní správu událostí a uživatelům poskytne snadný přístup k informacím o kulturních akcích. V závěrečné části budou navrženy možnosti dalšího rozvoje vytvořeného systému.

I. TEORETICKÁ ČÁST

1 SOUČASNÝ STAV PROBLEMATIKY

V případě, že se rozhodneme uspořádat vlastní kulturní akci nebo jinou veřejnou událost a je v našem zájmu ji představit široké veřejnosti, je nezbytné se rozhodnout jakým způsobem budeme akci propagovat. V dnešní době existuje mnoho různých platforem, ale i jiných řešení, které umožňují organizátorům propagaci a správu vlastních akcí.

Volba platformy pro propagaci kulturní akce či události závisí na charakteru akce. Je třeba zvážit, zda akce cílí na lokální komunitu nebo má ambice oslovit širší, možná dokonce mezinárodní publikum. K dispozici jsou specializované platformy určené pro konkrétní typy akcí, jako jsou hudební festivaly, výstavy nebo konference. Na druhou stranu existují i platformy, které jsou více univerzální a lze tak na nich propagovat různé typy akcí. Každá z těchto platforem nabízí různé nástroje, které následně organizátorům usnadňují práci se správou akcí.

Jednou z klíčových rolí v rozhodovacím procesu, jak akci propagovat je rozpočet. Organizátoři s nižším rozpočtem mohou využít nízkonákladové nebo bezplatné nástroje jako jsou například sociální média. V případě, že je rozpočet větší, mohou organizátoři uvažovat o využití marketingových agentur, které nabízí komplexní služby v oblasti propagace, včetně psaní PR článků, vytváření online reklam a podobně.

Dalším důležitým faktorem je velikost a dosah akce. Pro menší lokální události může být vhodnější využití regionálních medií, zatímco pro větší akce s nadregionálním nebo nadnárodním přesahem je efektivnější využití velmi známých platforem, které využívají miliony uživatelů.

Při rozhodování mezi využitím internetové platformy či specializované agentury pro propagaci kulturních akcí je klíčové posoudit jaké specifické potřeby a cíle akce má.

Platformy pro správu kulturních akcí nabízí širokou paletu nástrojů, které usnadňují pořadatelům správu a evidenci kulturních akcí. A to už od registrace účastníků až po prodej vstupenek a analýzu úspěšnosti akce. Zároveň mohou být tyto platformy efektivní i pro konkrétní uživatele, kteří se díky nim mohou dozvědět o akcích ve svém okolí a jednoduše je mohou sdílet.

Tyto nástroje jsou ideální pro organizátory, kteří preferují přímou kontrolu nad každým aspektem akce a hledají cenově dostupné řešení s možností rychlého nasazení. Díky tomu, že mají nad akcí přímou kontrolu se tak mohou dostat i do bližšího kontaktu s potenciálními

návštěvníky akce. Na druhé straně, specializované agentury nabízí komplexní služby, které v rámci propagace události zahrnují například marketingovou strategii či tvorbu obsahu. Tato řešení jsou vhodná pro organizátory s větším rozpočtem, kteří si přejí svěřit propagaci akce do rukou odborníků, aby se mohli plně soustředit na jiné aspekty přípravy události.

Při výběru mezi těmito dvěma přístupy musí organizátoři zvážit, zda je pro ně důležitější personalizace, kontrola a možnost přímé interakce s cílovou skupinou, či pohodlí a odborný přístup, kterou přináší spolupráce s agenturou.

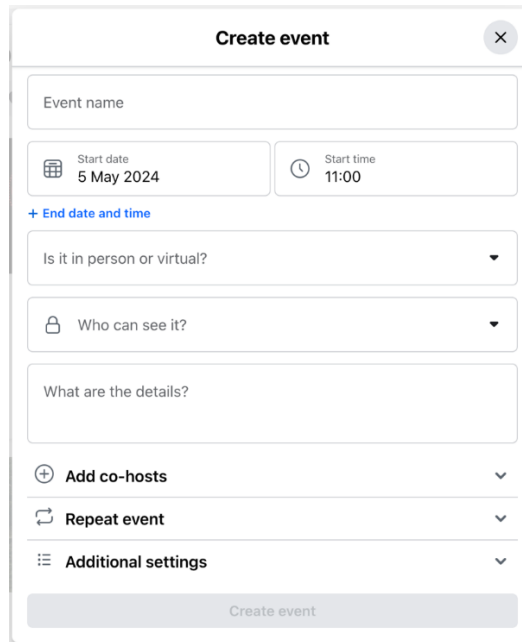
1.1 Facebook Events

Facebook, založený Markem Zuckerbergem a jeho spolužáky z Harvardovy univerzity v roce 2004, se rychle stal největší sociální sítí na světě. Jeho popularita postupně narostla díky inovativním funkcím, včetně Facebook Events. Tento nástroj umožňuje vytváření, sdílení a správu událostí, což umožňuje uživatelům propagovat nejrůznější akce, od soukromých setkání po velké veřejné akce. Facebook Events tak poskytuje efektivní způsob, jak zvýšit povědomí o plánované události a dosáhnout vyšší účasti na akcích. [1]

Události na Facebooku mohou obsahovat základní informace jako název akce, datum a čas, místo konání, jméno organizátora. Také může být doplněn popis akce, obrázek akce, kategorie akce a informace o vstupenkách. Místo konání akce se zobrazuje pomocí interaktivní mapy.

Facebook Events také nabízí možnost tvořit virtuální akce, které mohou využívat společnosti či samotní uživatelé. Tyto virtuální akce byly velmi efektivní v době pandemie Covid-19, kdy nebylo možné pořádat akce prezenčně. Pro virtuální akce je možné do popisu akce zahrnout i odkaz na online přenos.

U těchto akcí je jednoduché zjistit, kolik uživatelů má zájem o danou akci, popřípadě kolik uživatelů se plánuje akce zúčastnit.



Obrázek 1 Formulář pro vytvoření akce na platformě Facebook [2]

1.1.1 Výhody

1. Široký dosah: Díky obrovské uživatelské základně Facebooku mohou být akce snadno propagovány mezi široký okruh lidí.
2. Integrace se sociální sítí: Uživatelé mohou snadno sdílet události, popřípadě pozvat své přátele. Také mohou vidět, kdo z jejich přátel má o danou akci zájem nebo se jí plánuje zúčastnit.
3. Interaktivita: Organizátoři mohou komunikovat s účastníky a odpovídat na jejich dotazy. Popřípadě mohou aktualizovat informace o akci v reálném čase.
4. Zdarma: Vytvoření a sdílení akce na Facebooku je zdarma, což je obzvláště přínosné pro organizátory s omezeným rozpočtem.

1.1.2 Nevýhody

1. Přehlcení informacemi: Uživatelé mohou být zaplaveni velkým množstvím událostí, což může vést k tomu, že některé akce zůstanou bez povšimnutí.
2. Závislost na algoritmech: Úspěch akce může být silně závislý na algoritmech Facebooku, které určují, zda se událost objeví na hlavní stránce uživatelů, což může velmi omezit dosah sdělení.
3. Omezené možnosti designu: I když Facebook umožňuje určitou míru personalizace akcí, možnosti designu jsou i přesto omezenější ve srovnání s webovými stránkami, které se přímo zaměřují na evidenci akcí.

1.2 Kudy z nudy

Kudy z nudy je český turistický web, který poskytuje inspiraci a informace o cestování, volnočasových aktivitách a kulturních akcích po celé České republice. [3]

Stránka je zaměřena na širokou škálu uživatelů – od rodin s dětmi, přes milovníky historie a kultury až po adrenalinové nadšence. Slouží jako komplexní průvodce pro plánování volnočasových aktivit, výletů a objevování nových míst. [3]

Akce na Kudy z nudy může obsahovat základní informace jako je její název, datum a čas konání a místo konání s mapou. Dále je zde možné doplnit podrobný popis akce, obrázky, informace o vstupném, popřípadě kategorii akce. Také jsou zde kontaktní informace na organizátora, různá doporučení pro návštěvníky a odkazy na rezervaci nebo koupi vstupenek.

1.2.1 Výhody

1. Rozmanitost nabídky: Webová stránka pokrývá širokou škálu aktivit a zajímavostí z celé České republiky, není zaměřená pouze na časově omezené události. Uživatel snadno může spojit návštěvu akce s objevováním krás České republiky.
2. Uživatelsky přívětivé: Kudy z nudy umožňuje snadné vyhledávání dle kategorií, popřípadě lokality nebo data. Uživatel si tedy může přehledně filtrovat akce dle své potřeby.
3. Inspirace pro výlety: Nabízí nejen seznam akcí, ale i tipy na výlety, ubytování a stravování. Uživatel si může plně naplánovat nejen jednodenní výlet, ale i vícedenní dovolenou.

1.2.2 Nevýhody

1. Geografické pokrytí: Některé oblasti mohou být reprezentovány lépe než jiné, což může ovlivnit rozmanitost a dostupnost akcí.
2. Závislost na externích zdrojích: Informace mohou být částečně závislé na údajích poskytnutých třetími stranami, což může ovlivnit jejich přesnost.
3. Interakce s účastníky: Je zde menší prostor pro interakci mezi organizátory a účastníky. Organizátoři se o svých nedostatcích mohou dozvědět pouze z jiných platforem či vůbec. To samé platí pro uživatele, kteří zpětnou vazbu mohou vyjádřit pomocí jiných platforem nebo vůbec.

1.3 PRAGOKONCERT BOHEMIA, a. s.

PRAGOKONCERT BOHEMIA, a. s. je umělecká agentura působící v České republice, která má dlouhou historii úspěšného působení v oblasti organizace kulturních akcí. Agentura je jednou z nejstarších v zemi a vyznačuje se širokým spektrem aktivit, od pořádání festivalů a koncertů až po provozování hudebních klubů a zastupování umělců. [4]

Mezi vysoce uznávané festivaly společnosti PRAGOKONCERT BOHEMIA, a. s. patří Masters of Rock, Metalfest Open Air nebo Rock Castle. Kromě toho provozuje i kulturní prostory jako multikulturní víceúčelový sál Masters of Rock Café. [4]

Samotné akce na stránce PRAGOKONCERT BOHEMIA, a. s. obsahují základní informace ohledně názvu akce, datu a místa konání. Dále je stránka rozdělena na několik částí, jednou z nichž je kategorie vstupenky, kde si uživatel může přímo zakoupit vstupenky, popřípadě přes předprodejní síť. Následně je zde detail s popisem akce, program akce, pravidla akce a mapa, která zobrazuje místo konání.

1.3.1 Výhody

1. Široký rozsah služeb: PRAGOKONCERT BOHEMIA, a. s. nabízí komplexní služby pro organizaci kulturních akcí, včetně možnosti „akce na klíč“, které pokrývají vše od konceptu po realizaci. [5]
2. Bohaté zkušenosti: Dlouholetá historie a široké spektrum úspěšně realizovaných akcí jsou důkazem odbornosti a spolehlivosti agentury.
3. Prestiž a reputace: Spolupráce s renomovanou agenturou jako je PRAGOKONCERT BOHEMIA, a. s. může zvýšit prestiž akce.

1.3.2 Nevýhody

1. Cenová dostupnost: Vzhledem k rozsahu a kvalitě poskytovaných služeb mohou být náklady na spolupráci s PRAGOKONCERT BOHEMIA, a. s. vyšší a nemusí být tak vhodné pro organizátory akcí s nižším rozpočtem.
2. Zaměření na určité žánry: I když agentura pokrývá široké spektrum kulturních akcí, její hlavní zaměření je na rockové festivaly a jazz. Někteří uživatelé vyhledávající informace o akcích na této webové stránce se proto mohou cítit ochuzeni i o další akce.
3. Omezená flexibilita pro malé akce: Agentura je ideální pro velké a středně velké akce, nikoliv pro malé akce.

2 POUŽITÉ NÁSTROJE A TECHNOLOGIE

2.1 Vývojové prostředí

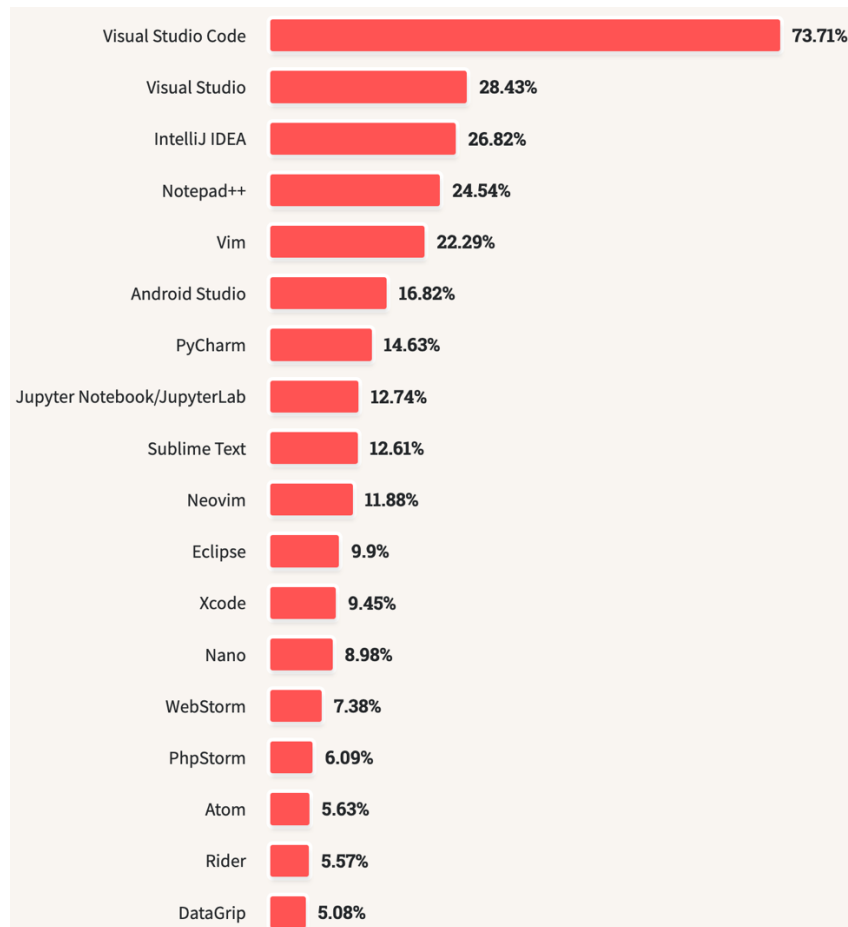
V moderním softwarovém inženýrství je používání integrovaných vývojových prostředí (IDE) nezbytností, která výrazně usnadňuje a zefektivňuje vývoj aplikací. IDE slouží jako nástroj pro programátory, nabízející komplexní sadu nástrojů v jedné unifikované softwarové aplikaci. [6]

Zahrnuje zejména editor kódu, nástroje pro kompilaci a ladění kódu, a často i další pomocné nástroje jako jsou nástroje pro správu verzí kódu a podporu pro vývojové frameworky. [6]

Používání IDE zvyšuje produktivitu vývojářů díky automatizaci opakujících se úkolů a poskytování inteligentních funkcí, jako je syntaktické zvýraznění a automatické doplňování kódu. [6]

Na trhu existuje řada IDE, z nichž každé je obvykle optimalizováno pro specifické programovací jazyky nebo platformy. Eclipse a IntelliJ IDEA jsou oblíbené mezi Java vývojáři, zatímco Visual Studio Code je univerzální IDE podporující mnoho jazyků a je oblíbené pro svou jednoduchost.

Z průzkumu popularity integrovaných vývojových prostředí (IDE) od Stack Overflow z roku 2023, je mezi respondenty nejpoužívanějším IDE Visual Studio Code s 73,71 %. Na druhém místě s 28,43 % skončilo IDE od stejného vývojáře, kterým je Visual Studio. Mezi další IDE patří IntelliJ IDEA, Notepad+ a Vim. [7]



Obrázek 2 Nejpoužívanější IDE v roce 2023 podle průzkumu Stack Overflow [7]

2.1.1 Visual Studio 2022

Visual Studio 2022 je populární vývojové prostředí od Microsoftu, které nabízí rozšířenou podporu pro mnoho programovacích jazyků, včetně C#, C++, a Visual Basic. Toto IDE je populární díky svým schopnostem napříč různými platformami, což zahrnuje vývoj desktopových, webových, cloudových a mobilních aplikací. Visual Studio 2022 poskytuje pokročilé nástroje pro ladění a testování kódu, integraci s cloudovou platformou Azure a podporuje efektivní správu zdrojového kódu prostřednictvím integrované podpory Git a GitHub. [8]

2.2 Databáze

Databáze je organizovaný systém pro ukládání, manipulaci a správu dat, který umožňuje vyhledávání a aktualizaci dat. Používá se v široké škále aplikací, od bankovních systémů až po osobní databáze. [9]

2.2.1 Relační databáze (SQL)

Koncept relačních databází poprvé představil v roce 1969 Edgar F. Coddem, vědec z IBM, ve svém průlomovém článku "A Relational Model of Data for Large Shared Data Banks". Coddův model zavedl myšlenku ukládání dat ve strukturovaných tabulkách a položil základy pro SQL, což revolučně změnilo způsob práce s databázemi. [10]

Relační databáze je typ databáze, který využívá relační model k ukládání a správě dat. V relačních databázích jsou data organizována do tabulek, které se skládají z řádků a sloupců. Každý řádek tabulky reprezentuje jednu instanci určitého objektu nebo vztahu a každý sloupec atribut tohoto objektu. Relační databáze umožňují definovat vztahy mezi tabulkami, což napomáhá zachovat integritu a konsistenci dat. [10]

Structured Query Language (SQL) je standardizovaný programovací jazyk určený pro správu relačních databází a provádění různých operací s daty v nich. SQL je široce využíván nejen databázovými administrátory, ale také vývojáři, kteří pomocí něj píšou skripty pro integraci systémů, a datovými analytiky, kteří ho používají pro sestavování a vykonávání analytických dotazů. Tento jazyk umožňuje upravovat struktury databázových tabulek a indexů, přidávat, aktualizovat, a mazat data v databázi. Mezi příklady RDBMS (systémy pro správu relačních databází) postavených na SQL patří Microsoft SQL Server, MySQL, PostgreSQL. [11]

Relace mezi tabulkami jsou základem relačních databází a umožňují definovat vztahy mezi různými datovými sady. Existují tři základní typy vztahů:

- Jeden k jednomu (1:1): Každý řádek v jedné tabulce je spojen s nejvýše jedním řádkem v jiné tabulce. [12]
- Jeden k mnoha (1:N): Jeden řádek v jedné tabulce může být spojen s více řádky v jiné tabulce. [12]
- Mnoho k mnoha (M:N): Více řádků v jedné tabulce může být spojeno s více řádky v jiné tabulce; tento vztah obvykle vyžaduje použití propojovací tabulky. [12]

Tyto relace jsou definovány pomocí klíčů – primárních klíčů, které jednoznačně identifikují řádek v tabulce, a cizích klíčů, které odkazují na primární klíč v jiné tabulce, čímž se vytváří vztah mezi tabulkami.

2.2.2 Nerelační databáze (NoSQL)

Nerelační databáze je typ databázového systému navržený pro správu a ukládání velkého množství nestructurovaných a polostucturovaných dat. Na rozdíl od tradičních relačních databází, které používají tabulky s předem definovanými schématy pro ukládání dat, nerelační databáze využívají flexibilní datové modely, které se mohou přizpůsobit změnám ve struktuře dat a jsou schopné horizontálního škálování pro zpracování rostoucího množství dat. [13]

Nerelační databáze se nejčastěji rozdělují do 4 kategorií:

- Dokumentově orientovaná databáze (například MongoDB nebo Cloudant);
- Databáze typu klíč-hodnota (například Memchaded nebo Redis);
- Sloupcově orientovaná databáze (například Hbase nebo Accumulo);
- Grafická databáze (například Amazon Neptune nebo Neo4j). [13]

2.2.3 Hierarchické databáze

Ukládají data v struktuře podobné stromu, kde každý záznam má jednoho rodiče. Tento typ byl běžně používán v minulosti a je přirozeně vhodný pro data, která dobře pasují do hierarchické struktury, jako jsou souborové systémy nebo organizační struktury. [14]

2.2.4 Síťové databáze

Oproti hierarchickým databázím umožňují složitější vztahy s více rodiči pro jeden záznam, což umožňuje modelování složitějších struktur než hierarchické databáze. Tyto databáze jsou reprezentovány pomocí grafu místo stromu. [14]

2.2.5 Objektově orientované databáze

Data jsou reprezentována jako objekty, což umožňuje integraci s objektově orientovanými programovacími jazyky. Tento přístup usnadňuje ukládání složitých datových struktur a podporuje koncepty objektově orientovaného programování. [15]

2.3 SQL Server Management Studio (SSMS)

SQL Server Management Studio (SSMS) je integrované prostředí pro správu SQL infrastruktury. Umožňuje uživatelům přístup, konfiguraci, správu a administraci různých SQL komponent, včetně SQL Serveru, Azure SQL Database, Azure SQL Managed Instance, SQL Server na Azure VM a Azure Synapse Analytics. SSMS kombinuje grafické nástroje

s editory skriptů, což poskytuje přístup k SQL Serveru pro vývojáře a databázové administrátory všech úrovní dovedností. [16]

Mezi nástroje zahrnuté v SSMS patří Object Explorer pro správu objektů v instancích SQL Serveru. Dále Template Explorer pro sestavování a správu šablon textů, Solution Explorer pro správu administrativních položek, Visual Database Tools pro sestavování dotazů a diagramů databází. [16]

2.4 HTML

HTML (Hypertext Markup Language) je základním jazykem používaným pro tvorbu a strukturování obsahu na internetu. Nejnovější a současně platná verze, HTML5, přináší řadu pokročilých funkcí, které vývojářům umožňují vytvářet bohatější a interaktivnější webové aplikace. HTML5 zahrnuje vylepšenou podporu pro multimédia, grafiku a formy interaktivních aplikací, což umožňuje vývojářům vytvářet složitější stránky bez závislosti na externích pluginech jako Adobe Flash. [17]

HTML dokument je založen na značkách (tags), které definují různé části a elementy stránky. Tyto značky jsou uspořádány hierarchicky a tvoří DOM (Document Object Model), což je struktura, kterou prohlížeče používají k interpretaci a zobrazení obsahu. [17]

Kód níže ukazuje základní strukturu HTML dokumentu, která zahrnuje deklaraci typu dokumentu, kořenový element `<html>`. Samotná HTML stránka obsahuje hlavičku dokumentu `<head>`, kde se primárně definují věci jako titulek stránky, odkazy na knihovny, skripty nebo metadata, která mohou definovat znakovou sadu dokumentu nebo třeba jméno autora webové stránky. Následuje `<body>`, které je samotné tělo stránky. Zde se již definují texty, odkazy, tabulky, obrázky a další věci, které se zobrazují na stránce. Na konci HTML kódu je patička `<footer>`, která obvykle obsahuje informace o autorovi stránky, copyright a další.

```
<!DOCTYPE html>
<html>
<head>
  <title>Moje první webová stránka</title>
</head>
<body>
  <h1>Vítejte na mé webové stránce</h1>
  <p>Toto je odstavec textu na webové stránce.</p>
  <a href="https://www.utb.cz/">Odkaz na stránky UTB</a>
</body>
<footer>
<p>Zápatí stránky</p>
</footer>
</html>
```

Obrázek 3 Ukázka HTML jazyka

2.5 CSS

CSS (Cascading Style Sheets) definuje, jak HTML elementy budou zobrazeny na obrazovce. CSS umožňuje webovým vývojářům oddělit obsah od designu a tím efektivně řídit layout, barvy, fonty a mnoho dalšího. [17]

Kód níže zobrazuje nastavení vzhledu elementů v <body> dokumentu, kde font-family určuje typ písma, background-color barvu pozadí a color barvu textu. Dále upravuje vzhled nadpisu <h1>, odstavce <p> a <div>, který se používá pro seskupování určitých elementů.

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  color: #333;
}

h1 {
  color: blue;
  margin-bottom: 20px;
}

p {
  line-height: 1.5;
}

div {
  text-align: center;
}
```

Obrázek 4 Ukázka CSS

2.6 JavaScript

JavaScript je univerzální skriptovací jazyk, který se používá pro přidávání interaktivních prvků na webové stránky. Tento jazyk umožňuje webovým vývojářům vytvářet dynamické uživatelské rozhraní, provádět asynchronní požadavky na server bez nutnosti obnovit stránku, manipulovat s DOM a vytvářet komplexní grafiku a animace. Umožňuje stránkám reagovat na uživatelské akce v reálném čase díky událostem jako je kliknutí myši, stisk klávesy, pohyb myši a další. [17]

V kódu níže je jednoduchý příklad použití JavaScriptu k interakci s HTML dokumentem. Kód obsahuje jednoduchou funkci, která mění text v odstavci na stránce, když uživatel klikne na tlačítko.

```
<!DOCTYPE html>
<html>
<head>
  <title>Interaktivní webová stránka</title>
</head>
<body>
  <h1>Vítejte na mé interaktivní webové stránce</h1>
  <p id="demo">Toto je odstavec čekající na interakci.</p>
  <button onclick="changeText()">Klikni na mě!</button>

  <script>
    function changeText() {
      document.getElementById("demo").innerHTML = "Text byl změněn!";
    }
  </script>
</body>
</html>
```

Obrázek 5 Ukázka JavaScript jazyka v HTML

2.7 ASP.NET CORE

ASP.NET Core je multiplatformní open-source webový framework vyvinutý společností Microsoft, který umožňuje vytvářet moderní, výkonné webové aplikace pro operační systémy Linux, macOS a Windows. [18]

2.7.1 C#

C# je moderní, objektově orientovaný programovací jazyk vyvinutý společností Microsoft, který se poprvé objevil v roce 2000 jako součást .NET Framework. Jazyk byl navržen Andersem Hejlsbergem a jeho týmem s cílem poskytnout vývojářům nástroj pro vývoj aplikací na platformě .NET. [19]

C# je silně typovaný jazyk, což znamená, že každá proměnná a objekt musí mít definovaný datový typ. Jednou z klíčových vlastností C# je jeho univerzálnost. Jazyk podporuje širokou škálu programovacích paradigmat, včetně imperativního, objektově orientovaného a funkcionálního programování. Díky této flexibilitě se C# stal populárním výběrem pro vývoj různých typů softwarů, od webových aplikací přes desktopové programy až po mobilní aplikace a hry. [19]

2.7.2 Blazor

Blazor je open-source webový framework, který je součástí ASP.NET Core ekosystému, a který umožňuje vývojářům psát klient- a server-side webové aplikace v jazyce C#. [20]

Framework Blazor využívá WebAssembly pro spouštění .NET kódu přímo v prohlížeči, což eliminuje potřebu používat JavaScript pro vytváření interaktivních uživatelských rozhraní. Tento přístup umožňuje vývojářům využít existující .NET knihovny a nástroje a poskytuje lepší výkon aplikací díky optimalizacím .NET runtime. [20]

Dále Blazor podporuje dva hlavní modely vývoje. Jedním z nich je Blazor Server, který běží na serveru a komunikuje s klientem pomocí SignalR. Další je Blazor WebAssembly, který umožňuje spouštění kódu přímo v prohlížeči uživatele. [20]

Obě varianty nabízí různé výhody, včetně snadné integrace s existujícími ASP.NET Core aplikacemi, možnosti opětovného použití kódu a vývoje v jazyce C#, který je vývojářům dobře známý.

```
<PageTitle>Counter</PageTitle>

<h1>Counter</h1>

<p role="status">Current count: @currentCount</p>

<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>

@code {
    private int currentCount = 0;

    private void IncrementCount()
    {
        currentCount++;
    }
}
```

Obrázek 6 Ukázka syntaxe Blazor [21]

2.7.3 Entity Framework Core

Entity Framework Core, často zkracovaný jako EF Core, je moderní, open-source ORM (Object-Relational Mapping) framework pro .NET. Jeho hlavním účelem je usnadnit vývojářům práci s databázemi tím, že umožňuje pracovat s daty pomocí objektových modelů .NET, namísto nutnosti psát SQL dotazy. EF Core překládá operace provedené na těchto objektech do SQL dotazů, které jsou kompatibilní s cílovou databází. Tím vývojářům umožňuje soustředit se na logiku aplikace bez nutnosti hluboké znalosti SQL. [20]

Jednou z klíčových vlastností EF Core je podpora LINQ (Language Integrated Query), která umožňuje vývojářům psát dotazy přímo v C# nebo jiných .NET jazycích. V kontextu EF Core, LINQ umožňuje vývojářům formátovat a provádět databázové dotazy pomocí syntaxe přirozené pro .NET jazyky, což zjednodušuje práci s daty a zvyšuje čitelnost kódu. [22]

Příkladem LINQ dotazu může být dotaz uvedený níže pro získání seznamu (listu) zeleniny z tabulky produktů. Metoda Where filtruje produkty podle typu, OrderByDescending zajišťuje sestupné řazení a ToList vrací výsledky jako seznam objektů.

```
var vegetables = context.Products
    .Where(p => p.ProductType.Contains("Vegetable"))
    .OrderByDescending(p => p.ProductName)
    .ToList();
```

Obrázek 7 Ukázka LINQ dotazu

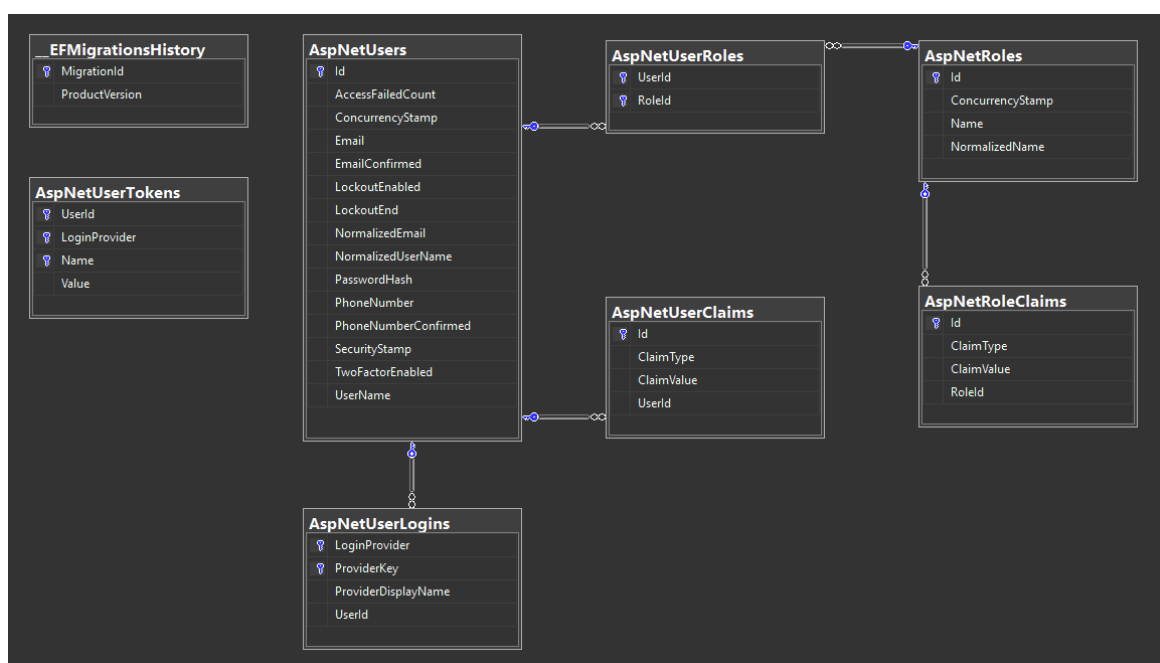
Přístupy při použití Entity Framework Core:

1. Code First: Tento přístup umožňuje vývojářům definovat databázové modely přímo v kódu pomocí tříd .NET. Databázové schéma je poté vygenerováno z těchto modelů. To poskytuje vysokou míru kontroly nad finálním schématem a je vhodné pro scénáře, kde je databáze navrhována souběžně s aplikací. [23]
2. Database First: V tomto přístupu začíná vývojář s existující databází. EF Core nástroje pak mohou být použity ke generování .NET tříd, které odpovídají databázovému schématu. Tento přístup je vhodný pro projekty, kde již databáze existuje a vývojáři chtějí vytvořit aplikaci, která s ní bude pracovat. [23]
3. Model First: Přístup Model First již v EF Core není přímo podporován tak, jak byl v předchozích verzích Entity Frameworku. V EF Core je doporučeným přístupem buď Code First nebo Database First, závisle na tom, zda vývoj začíná s databází nebo s modelem v kódu. [23]

2.7.4 Identity v ASP.NET Core

ASP.NET Core Identity je systém pro bezpečnou autorizaci a autentizaci uživatelů ve webových aplikacích postavených na ASP.Net Core. Tento komplexní framework poskytuje vývojářům sadu nástrojů pro správu uživatelských účtů, správu oprávnění a rolí a dalších funkcí. [24]

Jeho modulárnost umožňuje vývojářům integrovat a rozšiřovat standardní funkcionality podle specifických potřeb projektu, což zahrnuje i možnost přidávání vlastních polí do automaticky generovaných databázových tabulek pro uživatele.



Obrázek 8 Výchozí schéma tabulek vygenerovaných Identity [25]

Jedním z klíčových aspektů zabezpečení v ASP.NET Core Identity je ochrana hesel uživatelů. Framework využívá pokročilé metody hashování hesel, aby zabezpečil jejich uložení v databázi v nečitelné formě.

Výchozím hashovacím algoritmem v ASP.NET Core Identity je PBKDF2 (Password-Based Key Derivation Function 2). Tato funkce začíná kombinací vstupního hesla a náhodně generovaného saltu, což jsou náhodná data přidaná k heslu, aby i stejná hesla vyprodukovala různé hash hodnoty. Poté PBKDF2 používá hashovací algoritmus, jako je SHA-256 nebo SHA-512. Heslo a salt jsou opakovaně zpracovávány touto funkcí – počet opakování se nazývá iterace. Tím se zpomaluje proces hashování a ztěžuje tak případné útoky hrubou silou. Výsledkem PBKDF2 je odvozený klíč, který může být použit jako šifrovací klíč nebo uložen jako bezpečný hash hesla společně s použitým saltem. [26]

2.7.5 Návrhový vzor MVC

MVC neboli Model-View-Controller je architektonický vzor, který se používá zejména při vývoji webových aplikací. Původně byl vzor MVC používán pro grafická uživatelská rozhraní (GUI), jelikož vznikl dříve než samotné webové prohlížeče. [27]

Vzor MVC vznikl v 70. letech 20. století v rámci vývojových projektů v Xerox PARC, kde byl původně použit pro programování grafických uživatelských rozhraní v Smalltalku-80. Od té doby byl adaptován a rozšířen na různá programovací prostředí, včetně webu, kde dnes slouží jako základ pro mnoho moderních webových frameworků, jako jsou Ruby on Rails, Django (Python) a ASP.NET MVC. [27]

Samotná architektura se dělí na tři hlavní komponenty:

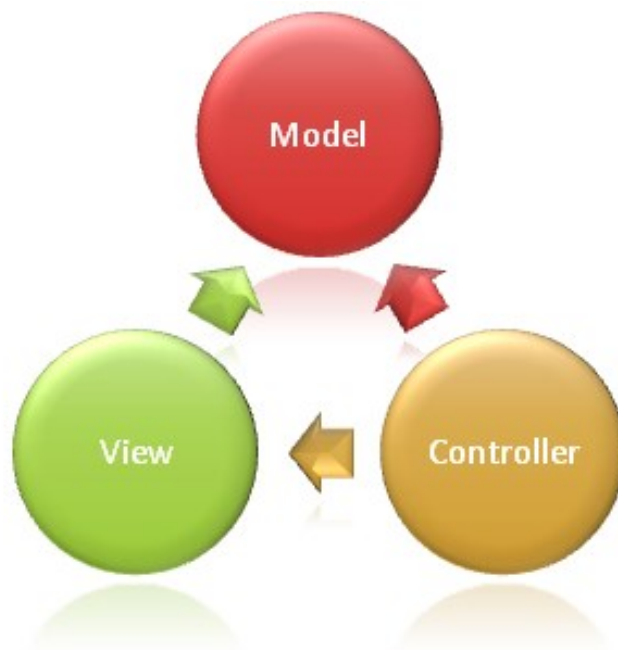
- Model – pracuje se samotnými daty. Model komunikuje s databází. [27]
- View (Pohled) – zobrazuje data na výstupu uživateli. Ve skutečnosti generuje uživatelské rozhraní. U webové aplikace pohled reprezentuje nejčastěji HTML, případně CSS. Pohled získává data, která jsou shromažďována komponentou modelu, ale tato data nejsou sbírána přímo, ale prostřednictvím ovladače, takže pohled mluví pouze k ovladači a neví odkud mu data přišla. [27]
- Controller – ovladač je prostředník, který propojuje model a pohled. Ovladač zpracuje uživatelský vstup a následně na něj reaguje tak, že vybere správný typ modelu. Model mu vrátí nějaká data, která upraví a pošle je do pohledu, který je následně zobrazí. [27]

Mezi hlavní výhody MVC patří:

- Oddělení zodpovědností: Tím, že MVC rozděluje aplikaci na tři hlavní komponenty, tak umožňuje vývojářům lepší správu kódu a jednodušší údržbu. [28]
- Usnadnění týmové spolupráce: Díky oddělení jednotlivých částí může na projektu pracovat více členů týmu bez vzájemného rušení. Backend programátoři mohou pracovat například na modelech, zatímco frontend programátoři se mohou zaměřit na programování pohledů. [28]
- Nezávislost komponent: Komponenty by měly být na sobě nezávislé, což znamená, že v případě změny jedné komponenty by nemělo dojít k ovlivnění ostatních komponent. [28]

Mezi hlavní nevýhody MVC patří:

- Složitost: Pro nováčky a programátorské začátečníky může být vzor MVC složitější na pochopení a následnou implementaci. Zejména pak u větších projektů s mnoha interakcemi mezi různými komponentami. [28]
- Duplikace kódu: V rozsáhlých aplikacích se můžeme setkat s duplicitním kódem, obzvláště když podobné datové operace nejsou mezi komponentami efektivně sdíleny, což může nastat při používání podobných procesů v různých částech aplikace. [28]



Obrázek 9 MVC architektura [29]

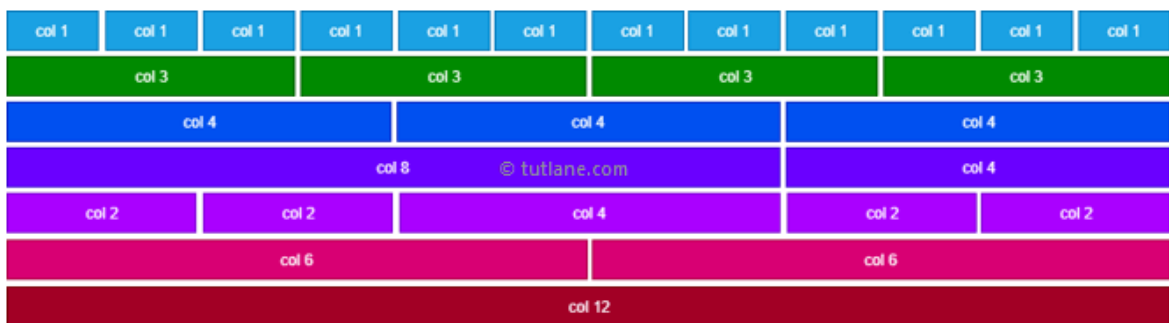
2.8 Bootstrap

Bootstrap je jedním z nejpoužívanějších front-endových frameworků pro vývoj responzivních a mobilně přizpůsobených webových stránek a aplikací.

Jde o open-source nástroj, který byl původně vyvinutý a uvedený v roce 2011 týmem ve společnosti Původně nazvaný Twitter Blueprint, byl vytvořen jako interní nástroj pro konzistentnost v projektování na Twitteru. S rostoucí potřebou unifikovaného designového jazyka a usnadnění práce vývojářům byl Bootstrap uvolněn jako open-source projekt, což umožnilo jeho další rozvoj komunitou dobrovolníků. Aktuálně nejnovější verzí Bootstrapu je verze 5. [30]

Díky své jednoduché použitelnosti a široké podpoře komunity se Bootstrap stal základním kamenem moderního webdesignu a vývoje webových aplikací. Jeho schopnost rychlého vývoje responzivních webových stránek z něj dělá vyhledávaný nástroj pro front-end vývojáře převážně z řad začátečníků.

Hlavní výhodou Bootstrapu je jeho systém mřížky (grid system), který umožňuje snadné rozdělení stránky do sloupců a řádků. Tento systém je klíčový pro vytváření responzivních layoutů, které se automaticky přizpůsobují velikosti obrazovky zařízení, na které jsou zobrazeny, ať už jde o stolní počítač, tablet, nebo mobilní telefon. Webové stránky jsou tak responzivní a snadno použitelné na širokém spektru zařízení bez potřeby psát pro každé zařízení specifický kód. [31]



Obrázek 10 Grid systém frameworku Bootstrap [32]

Bootstrap také obsahuje rozsáhlou sbírku předdefinovaných tříd a komponent, jako jsou tlačítka, navigační panely, formuláře, karty, modální okna a mnoho dalších, které lze snadno integrovat do webových stránek. Tyto komponenty nejenže zrychlují vývoj tým, že eliminují potřebu psát opakovaně stejný kód, ale také zajišťují konzistentní vizuální styl.

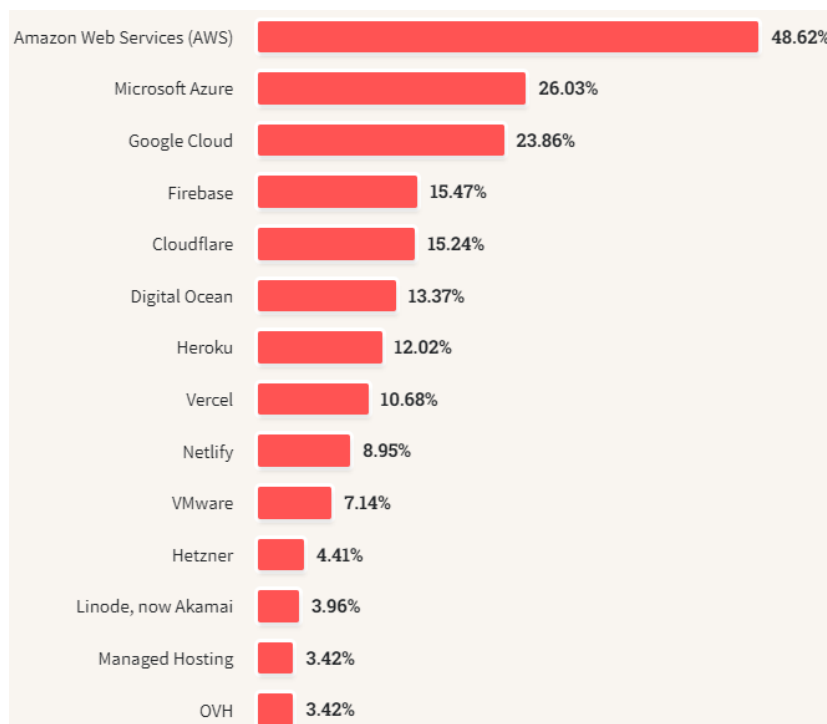
I přes spoustu výhod má Bootstrap i určité nevýhody. Jednou z nich může být to, že díky svým předdefinovaným komponentám se může stát, že webové stránky z vizuální strany nebudou tak originální a budou se podobat mnoha dalším stránkám, které využívají stejný framework. [33]

Dalším problémem může být pomalejší načítání jednotlivých stránek z důvodu načítání spousty CSS stylů a komponent, které ani nebudou využity. Nevýhodou je také horší customizace komponent s pomocí CSS. V rámci některých vlastností je možné narazit na problém, kdy je potřeba vlastnost přepsat z výchozí pomocí `!important`. [33]

2.9 Azure

Microsoft Azure je rozsáhlá cloudová platforma, která poskytuje celou řadu cloudových služeb umožňující firmám a vývojářům spravovat a rozšiřovat své aplikace. Jako jeden z předních hráčů na trhu cloudových služeb, Azure nabízí širokou škálu produktů a služeb včetně IaaS (Infrastructure as a Service), PaaS (Platform as a Service) a SaaS (Software as a Service) řešení, které podporují různé operační systémy, databáze, nástroje, programovací jazyky a frameworky. [34]

Hlavními konkurenty Azure jsou Amazon Web Services (AWS) a Google Cloud Platform (GCP). AWS, který je tržním lídrem, nabízí podobně rozsáhlý soubor služeb a je známý svou spolehlivostí a inovacemi v oblasti cloud computingu. [7]



Obrázek 11 Nejpoužívanější cloudové platformy v roce 2023 podle průzkumu Stack Overflow [7]

Google Cloud Platform vyniká v oblasti zpracování velkého množství dat a analytických služeb. Také je známý otevřeným cloudovým prostředím, které oproti Azure lépe usnadňuje integraci s dalšími technologiemi a službami převážně od společnosti Google. [35]

Azure účtuje své služby formou předplatného, které se odvíjí od rozsahu využitých zdrojů a služeb. Kromě toho Azure poskytuje také bezplatné předplatné pro některé služby, což umožňuje nasazení jednoduchých webových aplikací pro prezentaci a testování. [36]

V rámci široké nabídky produktů a služeb Azure se několik z nich vyčleňuje jako zásadní pro vývoj a hostování webových aplikací. Jedná se o Azure App Service, Azure SQL Database a Úložiště Azure (Azure Storage).

2.9.1 Azure App Service

Azure App Service je plně spravovaná platforma pro vývoj, nasazení a škálování webových aplikací. Podporuje širokou škálu programovacích jazyků jako jsou .NET, .NET Core, Java, Ruby, Node.js, PHP, nebo Python. [37]

Umožňuje vývojářům snadno nasazovat a spravovat webové aplikace bez nutnosti spravovat infrastrukturu. Nabízí integrované služby pro automatické škálování, zabezpečení a správu závislostí. [37]

2.9.2 Azure SQL Database

Azure SQL Database je relační databázová služba založená na technologii Microsoft SQL Server, ale plně spravovaná v cloudu. Tato služba poskytuje dynamické škálování kapacity, automatické zálohování, vysokou dostupnost a bezpečnost bez nutnosti fyzické správy serverů. Díky tomu je ideální pro aplikace vyžadující robustní databázové řešení s minimální administrativní námahou. [34]

2.9.3 Úložiště Azure (Azure Storage)

Azure Storage od společnosti Microsoft je robustní cloudové řešení pro ukládání dat, které nabízí vysokou odolnost, škálovatelnost a redundanci. Podporuje různé typy dat, včetně neřízených dat jako jsou obrázky a videa, log soubory, nebo velká datová úložiště pro aplikace. Každé úložiště dokáže pojmout až 500 TB dat. [34]

Úložiště je dostupné ve čtyřech variantách:

- Blob úložiště – služba pro ukládání velkého množství nestrukturovaných dat, jako jsou obrázky a videa, ve formě binárních objektů, nebo „blobů“;
- File úložiště – souborové úložiště přístupné pomocí standardních protokolů jako SMB;
- Queue úložiště – slouží k ukládání velkého počtu zpráv;
- Table úložiště – noSQL databázová služba pro ukládání a přístupu k velkému množství strukturovaných dat. [34]

3 POUŽITÉ KNIHOVNY

3.1 QuillJS

Pro tvorbu formátovaného textu na webových stránkách a v aplikacích se často využívají tzv. WYSIWYG editory. Zkratka WYSIWYG znamená "What You See Is What You Get", což v překladu znamená: „Co vidíte, to dostanete“. Tyto editory umožňují uživatelům snadno a intuitivně upravovat text a jeho formátování. Lze například měnit barvu, velikost písma, typ písma a další vlastnosti. [38]

QuillJS je příkladem moderního WYSIWYG editoru, který je navržen s důrazem na modularitu a rozšiřitelnost. To umožňuje vývojářům přizpůsobit editor přesně podle potřeb jejich konkrétního projektu, ať už jde o přidávání nových funkcí nebo odstraňování těch stávajících. [39]

Jedná se o open-source knihovnu, která má aktivní komunitu vývojářů, kteří se podílejí na jejím neustálém zdokonalování a rozšiřování o další funkce. [39]

3.2 Fullcalendar

FullCalendar je javascriptová knihovna pro vytváření responzivních interaktivních kalendářů na webových stránkách a v aplikacích. Umožňuje vývojářům snadno integrovat funkční kalendáře, které podporují různé zobrazení událostí za určité období. [40]

Díky široké škále konfiguračních možností lze kalendář přizpůsobit pro potřeby konkrétního projektu od jednoduchého zobrazení událostí po komplexní plánovací systémy.

3.3 Chart.js

Chart.js je aktuálně nejpopulárnější open-source javascriptová knihovna určená pro vizualizaci dat pomocí responzivních a interaktivních grafů. Pomocí této knihovny lze data zobrazovat v několika různých typech grafů, jako jsou čárové, sloupcové, koláčové a mnoho dalších. Typy grafů je možné mezi sebou kombinovat a vytvořit tak i různé typy smíšených grafů. [41]

Knihovna poskytuje bohaté možnosti konfigurace pomocí vlastních pluginů, které umožňují integrovat drag & drop, přiblížení, poznámky apod. [41]

3.4 Itext7

iText7 je open-source knihovna pro práci s PDF dokumenty v jazyce Java a .NET, která nabízí komplexní sadu funkcí a nástrojů pro manipulaci s PDF soubory. Jednou z hlavních výhod této knihovny je její široká škála možností, jak pracovat s PDF dokumenty. [42]

S využitím iText7 lze snadno vytvářet nové PDF dokumenty, editovat existující dokumenty včetně možnosti jejich spojování nebo rozdělování a také provádění různých operací, jako je přidávání nebo odebrání stránek, vkládání textu, obrázků či formulářových prvků. Tato knihovna rovněž umožňuje extrakci textu, obrázků a dalších dat z PDF dokumentů pro další zpracování.

Díky své modulární architektuře je iText7 velmi flexibilní a přizpůsobitelný, což umožňuje vývojářům integrovat pouze ty části knihovny, které potřebují, a tak minimalizovat velikost výsledné aplikace a zvýšit její výkon. Tato modularita také usnadňuje údržbu a rozšíření aplikací v průběhu času. Příkladem jedné z knihoven je „iText7.pdfHTML“, díky které lze vytvořit PDF soubor z HTML a CSS vstupu. [43][44]

Mezi další významné vlastnosti iText7 patří podpora šifrování a digitálních podpisů, což umožňuje zabezpečit PDF dokumenty proti neoprávněnému přístupu a zajištění autentičnosti a integrity jejich obsahu. Díky těmto vlastnostem je iText7 široce využíván v různých odvětvích, jako je správa dokumentů, reportování, automatizace procesů a mnoho dalších. [44]

4 TESTOVÁNÍ

Testování webových stránek je proces ověřování, zda web splňuje specifikované technické požadavky a očekávání uživatelů. Hlavním cílem je identifikovat chyby, problémy s používáním, nedostatky v bezpečnosti a další problémy před uvedením webu do provozu. Správné testování zajišťuje, že web je funkční, uživatelsky přívětivý, rychle reaguje a je kompatibilní s různými zařízeními a prohlížeči.

Testování lze rozdělit do mnoha skupin. Například podle znalosti kódu na black box, white box a gray box testy, podle způsobu realizace na manuální a automatizované nebo na funkční a nefunkční testování.

4.1 Druhy testování podle znalosti kódu

4.1.1 White box testování

White box testování je proces testování se znalostí kódu softwaru. Díky tomuto přístupu lze testovat algoritmy a identifikovat chyby přímo v konkrétních funkcích softwaru. Tento typ testování bývá časově náročnější než jiné typy. Příkladem takového testování mohou být unit testy. [45]

4.1.2 Black box testování

Black box testování je proces testování bez znalosti kódu softwaru. Tester pracuje pouze s grafickým rozhraním, kde testuje, zda software splňuje definované funkční požadavky. Tento typ testování je užitečný pro identifikaci chyb ve funkčnosti, které by mohly být pro uživatele viditelné. Příkladem takového testování jsou funkční nebo akceptační testy. [45]

4.1.3 Gray box testování

Gray box testování je hybridní přístup, který kombinuje prvky white box i black box testování. Počítá se s tím, že tester má částečnou znalost o vnitřní struktuře softwaru, ale primárně se zaměřuje na uživatelskou část. Časově je méně náročné než white box testování, ale poskytuje lepší možnosti pro identifikaci chyb než black box testy, protože testeři mají o softwaru více informací. [45]

4.2 Druhy testování podle úrovně

4.2.1 Unit testy (jednotkové testování)

Unit testy se zaměřují na ověření jednotlivých modulů aplikace izolovaně od zbytku systému. Cílem je zajistit, že každá funkce, metoda nebo třída, správně funguje na základě definovaných požadavků. Tento typ testování je prováděn vývojáři a je považován za první úroveň testování v procesu vývoje softwaru. Díky unit testům lze chyby odhalit již v rané fázi vývoje, což může výrazně snížit náklady na opravy chyb. [46]

4.2.2 Integroční testování

Integroční testování sleduje, jak různé jednotky spolupracují jako celek. Cílem je odhalit chyby ve vzájemném propojení jednotlivých modulů nebo funkcí aplikace. I když jednotky mohou fungovat správně samostatně, mohou při integraci nastat problémy, které tento typ testování odhaluje. Obvykle jsou prováděny po unit testech a mohou být prováděny nezávislými testery nebo samotnými vývojáři. [46]

4.2.3 Systémové testování

Systémové testování je komplexní testování celé aplikace v prostředí, které velmi připomíná reálné užívání produktu. Testuje, zda systém splňuje všechny specifikované požadavky a zda odpovídá kvalitativním standardům. Provádí se po integročním testování a obvykle se na něm podílejí nezávislí testéři, kteří nebyli zapojeni do vývoje programu. [46]

4.2.4 Akceptační testování

Akceptační testování je poslední fází testování, kde koneční uživatelé ověřují, zda systém odpovídá jejich potřebám a zda je připraven k nasazení. Tento test je zaměřen na to, jak dobře software plní požadované úkoly v reálném provozu a zda je akceptovatelný pro jeho plánované použití. [46]

II. PRAKTICKÁ ČÁST

5 ANALÝZA POŽADAVKŮ

Analýza požadavků je základním krokem v procesu vývoje softwaru, který pomáhá definovat, co software musí vykonávat a jaké podmínky musí splňovat, aby byl úspěšný. Rozdělení požadavků na funkční a nefunkční je klíčové pro úplné a jasné pochopení očekávání vůči systému.

Funkční požadavky specifikují konkrétní chování nebo funkce systému. Jedná se o akce, které systém musí vykonávat, jako například umožnění uživatelům se registrovat, přihlásit, měnit kontaktní údaje nebo vytvářet kulturní akce. Tyto požadavky jsou přímými ukazateli toho, co software musí umět a jak by měl uživatelům umožnit s ním interagovat.

Nefunkční požadavky oproti tomu definují kritéria, podle kterých se hodnotí systém, ale nespécifikují konkrétní chování. Tyto požadavky zahrnují nároky na bezpečnost, výkon, dostupnost, škálovatelnost a uživatelskou přívětivost. Jsou to atributy, které zvyšují uživatelskou spokojenost a zajišťují, že systém bude efektivně a bezpečně fungovat.

5.1 Funkční požadavky

1. Přihlášení a registrace

- 1) Uživatel se může registrovat do systému.
- 2) Uživatel se může přihlásit do systému.

2. Uživatel

- 1) Uživatel může změnit svou e-mailovou adresu.
- 2) Uživatel může změnit své heslo.
- 3) Uživatel může změnit své kontaktní údaje.

3. Společnost

- 1) Uživatel může vytvářet nové společnosti.
- 2) Admin může přidat uživatele jako nové členy do společnosti.
- 3) Admin může spravovat členy společnosti.
- 4) Admin může změnit kontaktní údaje společnosti.

4. Kulturní akce

- 1) Přihlášený uživatel může vytvořit vlastní kulturní akci, kde bude figurovat jako vlastník akce.
- 2) Uživatel společnosti může vytvořit kulturní akci, u které bude jako vlastník daná společnost.

- 3) Vlastník akce (uživatel/uživatel společnosti s právem Admin nebo Manager) může editovat kulturní akci.
 - 4) Vlastník akce (uživatel/uživatel společnosti s právem Admin nebo Manager) může mazat kulturní akci.
 - 5) Vlastník akce (uživatel/uživatel společnosti s právem Admin nebo Manager) může vytvořit závěrečnou zprávu.
 - 6) Přihlášený uživatel může zakliknout, že má zájem/chce se zúčastnit akce.
 - 7) Vlastník akce si může zobrazit statistiky akce, kde vidí v grafu počet uživatelů, kteří projeví zájem nebo se chtějí zúčastnit akce v daném čase.
5. Kalendář akcí
- 1) Uživatel si může zobrazit kalendář akcí s termíny, kterých se chce zúčastnit nebo má zájem se zúčastnit.
 - 2) Vlastníci akcí si mohou v kalendáři zobrazit všechny kulturní akce, jimiž jsou vlastníky.

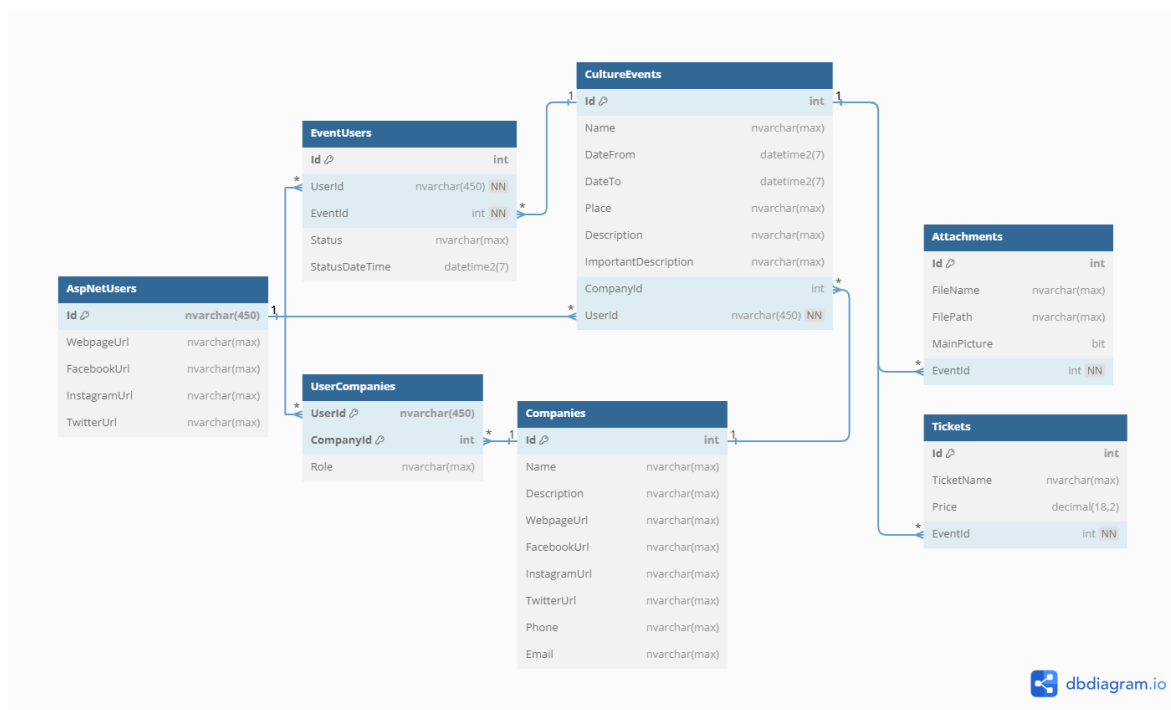
5.2 Nefunkční požadavky

1. Systém umožňuje autentizaci a autorizaci uživatelů na základě daných pravidel a rolí.
2. Pro správné fungování systému je potřebné připojení k internetu.
3. Heslo uživatele musí splňovat daný formát a je řádně zabezpečeno.
4. Systém by měl podporovat responzivní design aplikace na různých zařízeních.
5. Systém bude mít pochopitelné a uživatelsky přívětivé prostředí.
6. Pro správné fungování systému je nutný přístup k databázi.
7. Systém je schopen obsloužit více zdrojů dat zároveň.
8. Systém poskytne odpověď na požadavek v co nejkratším čase.

6 NÁVRH A DATABÁZE

Návrh databáze je důležitým krokem při vývoji jakékoliv webové aplikace, která bude interagovat s databází. Správný návrh nejenže zajišťuje efektivní ukládání a zpracování dat, ale také výrazně ovlivňuje celkovou škálovatelnost aplikace. Dobře strukturovaná databáze umožňuje snadnější správu dat a může zjednodušit používané databázové dotazy, které pak mohou mít vliv na rychlost odezvy celé aplikace. Další výhodou návrhu databáze je lepší představa o tom, jak samotný systém bude fungovat. To následně může pomoci v předvídání možných problémů, které by mohly během vývoje a fungování aplikace nastat.

Cílem této kapitoly je objasnit, k jakým účelům jsou určeny jednotlivé tabulky v rámci aplikace Cultur4U.



Obrázek 12 Návrh databáze s relacemi bez Identity

6.1 Tabulka AspNetUsers

První z tabulek je tabulka AspNetUsers, která je součástí infrastruktury Identity v ASP.NET Core a slouží k ukládání základních informací o registrovaných uživateli. Při vygenerování obsahuje několik polí k ukládání dat jako je e-mailová adresa, hash hesla, a další.

Navíc je tato tabulka rozšířená o pole potřebné pro potřeby aplikace. Jedná se o pole pro ukládání URL adres profilů/stránek různých sociálních sítí.

6.2 Tabulka CulturuEvents

Dále se zde nachází tabulka CulturuEvents, která slouží k uchovávání údajů o jednotlivých kulturních akcích. Uchovává například název kulturní akce, popis, místo konání a další informace.

Kulturní akce jsou spojeny s uživatelem, který je vytvořil (UserId) nebo společností (CompanyId).

6.3 Tabulka EventUsers

Následně tabulka EventUsers zaznamenává vztahy mezi jednotlivými uživateli a kulturními akcemi. Slouží tak k vytváření statistik zájmu o danou akci v určitém čase.

6.4 Tabulka Tickets

Také se zde nachází tabulka Tickets, která slouží k ukládání informací o vstupenkách pro jednotlivé akce. Každá akce může mít několik druhů vstupenek s odlišnými názvy i cenami.

6.5 Tabulka UserCompanies

Tabulka UserCompanies je důležitá pro ukládání vztahů mezi jednotlivými uživateli a společnostmi včetně jejich role. Umožňuje tak definovat v jakých společnost je uživatel členem a jakou roli v rámci společnosti má.

6.6 Tabulka Attachments

Dále se zde nachází tabulka Attachments, která obsahuje informace o přílohách. Každá příloha má své Id, název souboru a cestu k místu, kde je soubor uložený. Následně je obohacena o informaci, zda se jedná o hlavní obrázek a cizí klíč, který propojuje přílohu s konkrétní kulturní akcí.

6.7 Tabulka Companies

Poslední je tabulka Companies, která slouží k ukládání informací o vytvořených společnostech. Tato tabulka obsahuje název společnosti, popis a následně několik polí umožňujících uložení URL adres na sociální síť, popřípadě webové stránky.

6.8 Implementace databáze

Implementace databáze v projektu byla realizována pomocí přístupu Code-First v Entity Framework (EF). Tento přístup umožňuje vývojářům definovat strukturu databáze přímo v kódu projektu skrze definované třídy, známé jako entity.

Každá tabulka v databázi je reprezentována entitou v jazyce C#, kde definované atributy třídy přímo odpovídají sloupcům v tabulce. Příkladem může být tabulka pro ukládání informací ohledně kulturních akcí. U každé vlastnosti lze definovat název sloupce současně s definováním datového typu vlastnosti a případně i nastavením, že atribut může obsahovat null a je tedy nepovinný. V případě, že je potřeba, aby vlastnost byla nepovinná, tak za datový typ musí být umístěn symbol „?“.

Oproti tomu, když je nutné propojit tabulku s jinou tabulkou, lze využít virtuální vlastnosti. Tyto vlastnosti nevytváří nové sloupce v tabulce, ale pouze umožňují dynamicky načítat data z dané tabulky za běhu. Tato data tak jsou načtena až poté, co jsou explicitně vyžádána.

Dále je možné definovat i konkrétní atributy u vlastností jako [StringLength(255)], [Key], [Required] a podobně.

```
public class CultureEvent
{
    Počet odkazů: 20
    public int Id { get; set; }
    Počet odkazů: 14
    public string Name { get; set; }
    Počet odkazů: 14
    public DateTime? DateFrom { get; set; }
    Počet odkazů: 14
    public DateTime? DateTo { get; set; }
    Počet odkazů: 8
    public string? Place { get; set; }
    Počet odkazů: 9
    public string? Description { get; set; }
    Počet odkazů: 5
    public string? ImportantDescription { get; set; }
    Počet odkazů: 17
    public int? CompanyId { get; set; }
    Počet odkazů: 6
    public string UserId { get; set; }

    Počet odkazů: 14
    public virtual Company Company { get; set; }
    Počet odkazů: 14
    public virtual ApplicationUser User { get; set; }
    Počet odkazů: 16
    public virtual ICollection<Attachment> Attachments { get; set; }
    Počet odkazů: 11
    public virtual ICollection<Ticket> Tickets { get; set; }
}
```

Obrázek 13 Ukázka entity CultureEvent

Dalším krokem je vytvoření a nakonfigurování třídy DbContext. Tato třída řídí interakce mezi databází a kódem. V tomto případě byla tato třída vytvořena při přidání Identity do projektu. Následně v DbContext byly definovány všechny tabulky, které je potřeba vytvořit.

Metoda OnModelCreating slouží ke specifickému nastavení konfigurace tabulek a vztahů mezi nimi. V případě Cultur4U jsou v metodě definovány cizí klíče mezi tabulkami a jejich vztahy many to many.

Případně je zde možné například definovat chování entit závislých na sobě v případě smazání jedné z entit. Tedy jak se zachová entita v případě, že navazující entita bude smazána.

```
public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
{
    Počet odkazů: 0
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
        : base(options)
    {
    }
    Počet odkazů: 11
    public DbSet<CultureEvent> CultureEvents { get; set; }
    Počet odkazů: 8
    public DbSet<Company> Companies { get; set; }
    Počet odkazů: 4
    public DbSet<Attachment> Attachments { get; set; }
    Počet odkazů: 1
    public DbSet<Ticket> Tickets { get; set; }
    Počet odkazů: 10
    public DbSet<UserCompany> UserCompanies { get; set; }
    Počet odkazů: 11
    public DbSet<EventUser> EventUsers { get; set; }
    Počet odkazů: 0
    protected override void OnModelCreating(ModelBuilder builder)
    {
        base.OnModelCreating(builder);
        builder.Entity<UserCompany>()
            .HasKey(uc => new { uc.UserId, uc.CompanyId });
        builder.Entity<UserCompany>()
            .HasOne(uc => uc.User)
            .WithMany(u => u.UserCompanies)
            .HasForeignKey(uc => uc.UserId);
        builder.Entity<UserCompany>()
            .HasOne(uc => uc.Company)
            .WithMany(c => c.UserCompanies)
            .HasForeignKey(uc => uc.CompanyId);
        builder.Entity<EventUser>()
            .HasOne(eu => eu.User)
            .WithMany()
            .HasForeignKey(eu => eu.UserId)
            .onDelete(DeleteBehavior.NoAction);
        builder.Entity<EventUser>()
            .HasOne(eu => eu.Event)
            .WithMany()
            .HasForeignKey(eu => eu.EventId)
            .onDelete(DeleteBehavior.Cascade);
    }
}
```

Obrázek 14 Ukázka implementace ApplicationDbContext

Následně už stačí pomocí příkazu „add-migration nazezMigrace“ vytvořit migraci. Migrace slouží k ukládání různých stavů, kdy došlo ke změnám v databázové struktuře.

V případě, že by došlo k nějakému problému, tak lze využít právě migrace k vrácení struktury databáze na poslední funkční. Po vytvoření funkční migrace ji stačí aplikovat pomocí „update-database.“

7 WEBOVÁ APLIKACE CULTUR4U

Webová aplikace je software, který je hostován na webserverech a je přístupný prostřednictvím internetového prohlížeče. Tyto aplikace umožňují uživatelům provádět různé činnosti online, od jednoduchého prohlížení obsahu až po složitější interakce v reálném čase.

Cílem této kapitoly je poskytnout návod, jak vypadá a funguje webová aplikace Cultur4U, a jakým způsobem jsou implementovány a naprogramovány jednotlivé funkcionality.

7.1 Navigace

Navigace na webové stránce je základním a jedním z nejdůležitějších prvků, který umožňuje uživatelům přecházet mezi různými částmi aplikace.

V rámci webové aplikace je navigace obvykle součástí layoutu, což je šablona, která se nazývá `_Layout.cshtml`. V layout souboru se definují opakující se prvky uživatelského rozhraní, jako je hlavička stránky, navigační lišta a patička.

Hlavní výhodou je, že se kód používá napříč několika různými stránkami. Kód se tak snadněji udržuje a je přehlednější.

V kódu navigace je využito několik `if` podmínek, které zajišťují, jaké možnosti navigace se uživateli zobrazují v závislosti na tom, zda je nebo není přihlášený.

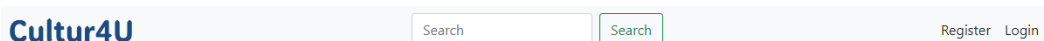
```

<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a style="padding-right:10px" asp-area="" asp-controller="Home" asp-action="Index">
      </a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
      aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        @if (SignInManager.IsSignedIn(User)) {
          <li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="Events" asp-action="EventList">Event List</a></li>
          <li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="Calendar" asp-action="Index">Calendar</a></li>
        }
      </ul>
      <div class="d-flex">
        <div class="position-relative">
          <input class="form-control me-2" id="searchQuery" name="searchQuery" type="search" placeholder="Search" aria-label="Search">
          <div id="searchResults" class="search-results"></div>
        </div>
        <button class="btn btn-outline-success" type="button" id="searchButton">Search</button>
      </div>
      <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
        @if (SignInManager.IsSignedIn(User)) {
          <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">
              @UserManager.GetUserName(User)
            </a>
            <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
              <li><a id="manage" class="dropdown-item" asp-area="Identity" asp-page="/Account/Manage/Index" title="Settings">Settings</a></li>
              <li><hr class="dropdown-divider"></li>
              <p style="font-size:10px; padding-left:10px; margin-bottom:0px; margin-left:10px">Companies</p>
              @await Component.InvokeAsync("CompaniesDropdown")
              <li><hr class="dropdown-divider"></li>
              <li><a id="manage" class="dropdown-item" asp-controller="Company" asp-action="Create" title="Create company">Create company</a></li>
              <li><hr class="dropdown-divider"></li>
              <li>
                <form id="logoutForm" class="form-inline" asp-area="Identity" asp-page="/Account/Logout"
                  asp-route-returnUrl="@Url.Action("Index", "Home", new { area = "" })">
                  <button id="logout" type="submit" class="dropdown-item">Logout</button>
                </form>
              </li>
            </ul>
          </li>
        }
        else {
          <li class="nav-item"><a class="nav-link text-dark" id="register" asp-area="Identity" asp-page="/Account/Register">Register</a></li>
          <li class="nav-item"><a class="nav-link text-dark" id="login" asp-area="Identity" asp-page="/Account/Login">Login</a></li>
        }
      </ul>
    </div>
  </div>
</nav>

```

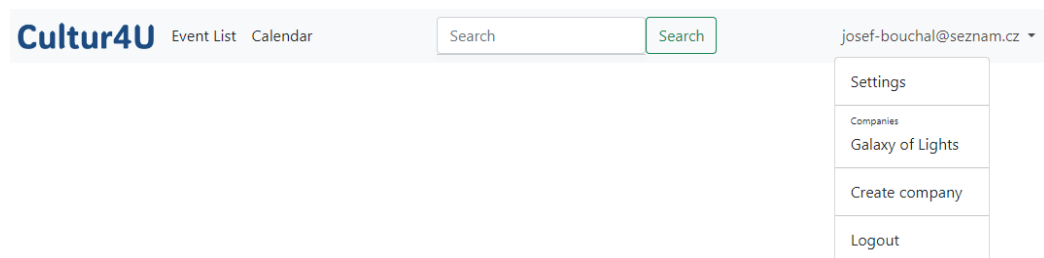
Obrázek 15 Implementace navigační lišty v _Layout.cshtml

Nepřihlášený uživatel v navigaci vidí logo aplikace s proklikem na domovskou stránku. Následně je zde vyhledávací pole pro vyhledávání konkrétních kulturních akcí a v poslední řadě tlačítka pro přihlášení a registraci uživatele.



Obrázek 16 Navigační lišta nepřihlášeného uživatele

U přihlášeného uživatele se navíc zpřístupní tlačítka Event List a Calendar. Dále místo tlačítek pro registraci/přihlášení se mu zobrazuje jeho e-mailová adresa s možností otevření dropdown seznamu. Tento dropdown seznam obsahuje tlačítka Settings pro proklik do nastavení uživatele. Následně je zde přehled všech společností, ve kterých je uživatel členem. A také je zde tlačítka pro vytvoření vlastní společnosti a pro odhlášení uživatele.



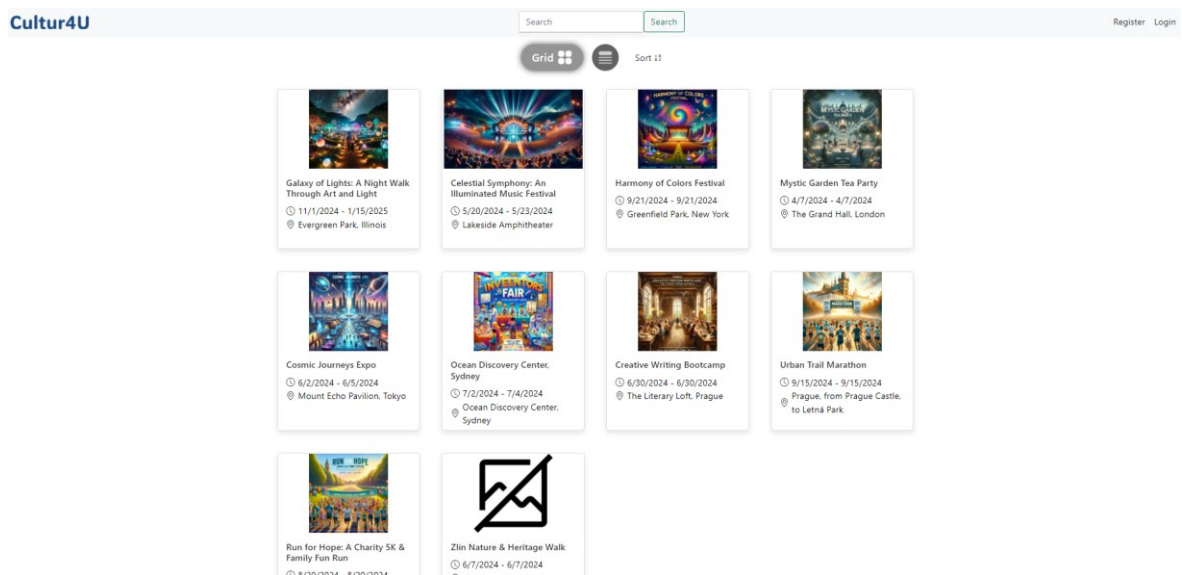
Obrázek 17 Navigační lišta přihlášeného uživatele

7.2 Domovská stránka

Domovská stránka je hlavní stránka webové aplikace, která slouží k zobrazení všech kulturních akcí. V rámci celé aplikace se jedná o jedinou stránku, která má stejný vzhled a obsah stránky pro přihlášené i nepřihlášené uživatele.

Akce jsou zobrazeny ve formě karet, přičemž každá karta obsahuje hlavní obrázek, název, datum a místo konání akce. Po přepnutí volby zobrazení na List se navíc zobrazuje i popis dané akce.

Kliknutím na jakoukoli kartu se uživatel dostane na detailní stránku akce, kde jsou dostupné podrobnější informace.



Obrázek 18 Domovská stránka

Zobrazení domovské stránky je řízeno pomocí kontroleru HomeController, který načítá data z databáze a předává je do pohledu. Tato data jsou načítána pomocí lambda výrazů.

V kódu níže (Obrázek 19) je zobrazeno, jak se využívá Entity Framework pro asynchronní načítání dat z databáze.

Využitím možnosti Select lze definovat jaké pole v tabulce bude předáno do view modelu, který následně chceme zobrazit v pohledu.

Po vytvoření listu eventů se použije ViewBag pro předání informace ohledně toho, zda se uživatel nachází na vývojové nebo produkční verzi webové stránky.

Tento proces je potřeba z toho důvodu, že cesta uložených příloh kulturních akcí je odlišná v obou verzích. Na vývojové verzi se soubory ukládají přímo do složky v projektu (kapitola 7.8.1.1), zatímco u produkční verze se soubory ukládají do Azure storage.

```
public async Task<IActionResult> Index()
{
    var events = await _context.CultureEvents
        .Select(e => new EventViewModel
        {
            Id = e.Id,
            EventName = e.Name,
            Description = e.Description,
            DateFrom = e.DateFrom,
            DateTo = e.DateTo,
            EventPlace = e.Place,
            ImagePath = e.Attachments.FirstOrDefault(a => a.MainPicture == true).FilePath
        }).ToListAsync();
    ViewBag.IsDevelopment = _env.IsDevelopment();

    return View(events);
}
```

Obrázek 19 Implementace metody Index

List kulturních akcí je následně předaný do pohledu, kde se v cyklu foreach vytvoří pro každou akci nová karta s atributem event-container. V rámci samotné karty se nachází podmínka if, která v první řadě zjišťuje, zda daná akce obsahuje odkaz v ImagePath.

Pokud kulturní akce obsahuje obrázek, tak je právě na základě další if podmínky kontrolováno, v jaké je uživateli verzi aplikace a podle toho se zobrazí obrázek se správnou cestou k danému souboru.

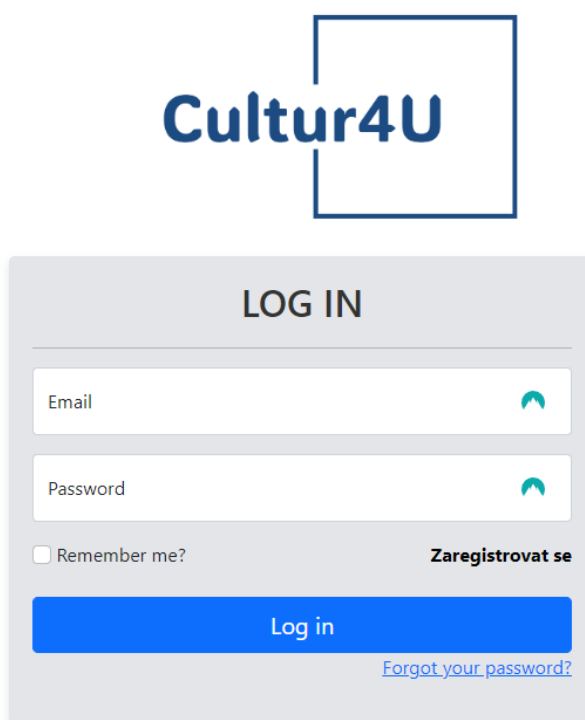
```
<div class="cards grid-view">
  @foreach (var eventVm in Model)
  {
    <div class="event-container">
      <a href="@Url.Action("EventDetail", "Events", new { id = eventVm.Id })" class="event-card-link">
        <div class="card" data-date-from="@eventVm.DateFrom">
          <div class="event-header">
            @if (!string.IsNullOrEmpty(eventVm.ImagePath))
            {
              if (ViewBag.IsDevelopment)
              {
                
              }
              else
              {
                
              }
            }
            else
            {
              
            }
          </div>
          <div class="event-details">
            <h2 class="event-name">@eventVm.EventName</h2>
            <div class="event-icons">
              <i class="bi bi-clock event-icon"></i>
              <span>@eventVm.DateFrom?.ToString("d") - @eventVm.DateTo?.ToString("d")</span>
            </div>
            <div class="event-icons">
              <i class="bi bi-geo-alt event-icon"></i>
              <span>@eventVm.EventPlace</span>
            </div>
            <div class="event-list-description">
              <p style="font-size: 1rem;font-weight:bold;margin: 5px 0px 0px 0px; border-top: 1px solid #lightgray">Description:</p>
              <p class="event-description">@HtmlUtilities.StripHtmlTags(eventVm.Description)</p>
            </div>
          </div>
        </div>
      </a>
    </div>
  }
</div>
```

Obrázek 20 Dynamické zobrazení akcí na hlavní stránce

7.3 Přihlášení

Formulář pro přihlašování i jeho fungování je vygenerovaný pomocí ASP.NET Core Identity a následně upravený pro potřeby webové aplikace. Samotné přihlášení do aplikace probíhá prostřednictvím zadání e-mailové adresy a hesla.

V případě, že uživatel zapomene heslo, tak si přes tlačítko „Forgot your password?“ může odeslat na svou e-mailovou adresu obnovovací odkaz pro vytvoření nového hesla.



The image shows a login form for Cultur4U. At the top, the Cultur4U logo is displayed. Below it is a grey box containing the login form. The form has a title 'LOG IN'. It features two input fields: 'Email' and 'Password', each with a green eye icon for visibility toggling. Below the password field is a checkbox labeled 'Remember me?' and a link 'Zaregistrovat se'. At the bottom of the form is a blue 'Log in' button and a link 'Forgot your password?'.

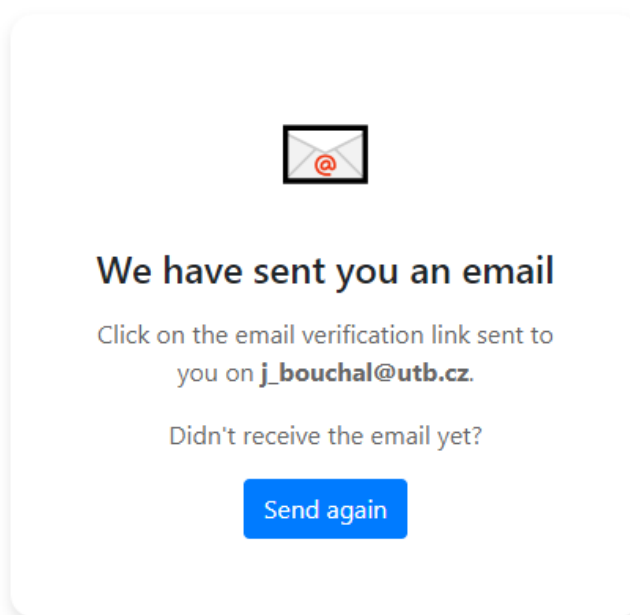
Obrázek 21 Formulář pro přihlášení do Cultur4U

7.4 Registrace

Formulář pro registraci je, stejně jako formulář pro přihlášení, generovaný pomocí ASP.NET Core Identity a následně upravený pro potřeby webové aplikace. V rámci registrace stačí zadat e-mailovou adresu, která ještě není registrovaná a heslo, které odpovídá formátu, který stránka požaduje. Při vyplňování hesla je povinné, aby obsahovalo minimálně 6 písmen, z toho alespoň 1 velké písmeno, 1 číslo a 1 znak/symbol.

Po registraci je uživatel přesměrován na potvrzovací obrazovku, kde je informován, že mu byl odeslán e-mail s verifikačním kódem pro potvrzení e-mailové adresy. Zároveň v případě chyby při neodeslání e-mailové zprávy je na obrazovce tlačítko pro opětovné odeslání

verifikačního e-mailu. Dokud uživatel e-mail nepotvrdí, tak je jeho účet neaktivní a nemůže se tak pomocí přihlašovacích údajů přihlásit.



Obrázek 22 Verifikační obrazovka

7.4.1 Odesílání verifikačního kódu

Při registraci nového uživatele je nezbytné ověřit jeho e-mailovou adresu, aby bylo zajištěno, že uživatel má skutečně přístup k e-mailové adrese, kterou zadal během registrace.

Pro odesílání verifikačních e-mailů byla implementována služba EmailSender, která využívá SMTP protokol pro komunikaci s e-mailovým serverem.

Aby tato služba fungovala správně, bylo nutné vytvořit e-mailovou adresu, kterou lze využít pro odesílání e-mailových zpráv. Pro účely této bakalářské práce byl vytvořen bezplatný Gmail účet (cultur4u@gmail.com).

EmailSender obsahuje metodu SendEmailAsync, která na základě vstupních parametrů jako je cílová e-mailová adresa, předmět zprávy a samotný obsah zprávy odešle e-mailovou zprávu.

Přihlašovací údaje do e-mailového účtu nejsou uloženy přímo ve vytvořené službě, protože je to bezpečnostní riziko. Místo toho jsou údaje uloženy v souboru appsettings.json.

Pro ještě vyšší úroveň zabezpečení a primárně v ostrém provozu je doporučeno využít služby jako je Azure Key Vault, které umožňují centralizované spravování a šifrování citlivých údajů. [47]

```
public class EmailSender : IEmailSender
{
    private readonly string _email;
    private readonly string _password;
    private readonly string _smtp;

    Počet odkazů: 0
    public EmailSender(IOptions<EmailConfiguration> options)
    {
        _email = options.Value.Email;
        _password = options.Value.Password;
        _smtp = options.Value.Smtp;
    }
    Počet odkazů: 0
    public async Task SendEmailAsync(string email, string subject, string htmlMessage)
    {
        var smtpClient = new SmtpClient(_smtp)
        {
            Port = 587,
            UseDefaultCredentials = false,
            Credentials = new NetworkCredential(_email, _password),
            EnableSsl = true
        };

        var mailMessage = new MailMessage
        {
            From = new MailAddress(_email),
            Subject = subject,
            Body = htmlMessage,
            IsBodyHtml = true
        };
        mailMessage.To.Add(email);

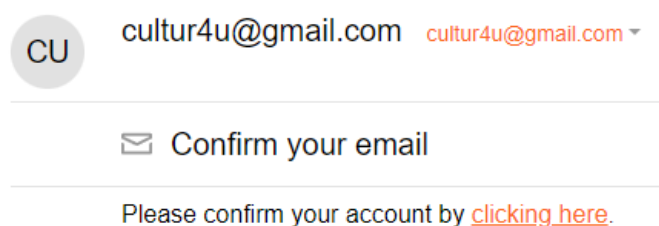
        await smtpClient.SendMailAsync(mailMessage);
    }
}

Počet odkazů: 2
public class EmailConfiguration
{
    Počet odkazů: 1
    public string Email { get; set; }
    Počet odkazů: 1
    public string Password { get; set; }
    Počet odkazů: 1
    public string Smtp { get; set; }
}
```

Obrázek 23 Ukázka implementace služby EmailSender

Po úspěšném odeslání e-mailové zprávy obdrží uživatel e-mail obsahující verifikační odkaz. Kliknutím na tento odkaz je uživatel přesměrován na stránku aplikace, kde je jeho e-mailová adresa potvrzena a účet aktivován.

Tento krok dokončuje proces registrace a zajišťuje, že e-mailová adresa je platná a aktivní.



Obrázek 24 Ukázka příchozí verifikační e-mailové zprávy

7.5 Detail akce

Detail akce je zobrazení konkrétní akce, zahrnující veškeré údaje o dané akci. Stránka je rozdělená na dva sloupce, kde se v levém sloupci zobrazují nejdůležitější informace o akci jako je název akce, datum a místo konání, možnosti vstupenek, kontaktní údaje pořadatele.

V pravém sloupci se zobrazuje hlavní obrázek akce spolu s celým popisem dané akce. Následně je zde také fotogalerie a videogalerie v případě, že akce tyto soubory obsahuje.

Po kliknutí na obrázek z fotogalerie se otevře jednoduché okno, kde uživatel může proklikávat mezi jednotlivými obrázky a vidí u nich i jejich názvy.

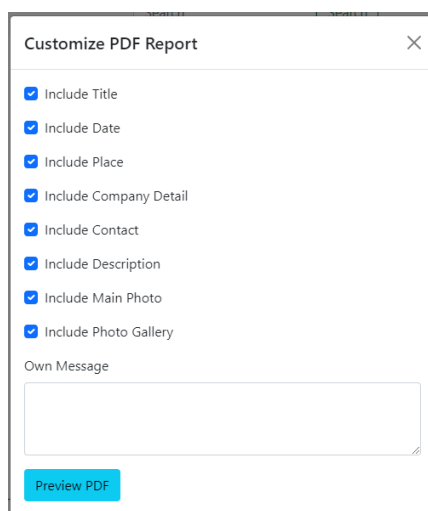
Přihlášení uživatelé, kteří nejsou organizátory dané akce navíc vidí v detailu akce tlačítka „I will attend“ a „I am interested“. Díky těmto tlačítkům mohou uživatelé projevit zájem o tuto kulturní akci nebo rovnou potvrdit svou účast.

Díky tomuto následně uživatelé vidí akce ve svém kalendáři (kapitola 7.9) a organizátorům to poskytuje přehled statistiky zájmu o konkrétní akci (kapitola 7.8.3).

Organizátoři akce v detailu akce vidí navíc tlačítko „Customize PDF Report“, pomocí kterého si mohou vygenerovat závěrečnou zprávu k dané události. (Příloha P I)

7.5.1 Tvorba závěrečné zprávy

Po kliknutí na tlačítko „Customize PDF Report“ se organizátorovi akce zobrazí jednoduchý formulář, kde si může definovat, které údaje o akci chce zobrazit v závěrečné zprávě. Zároveň zde má možnost dopsat na konec závěrečné zprávy svůj vlastní text, kde může případně zhodnotit spokojenost s akcí, uvést poděkování a podobně.



Customize PDF Report

- Include Title
- Include Date
- Include Place
- Include Company Detail
- Include Contact
- Include Description
- Include Main Photo
- Include Photo Gallery

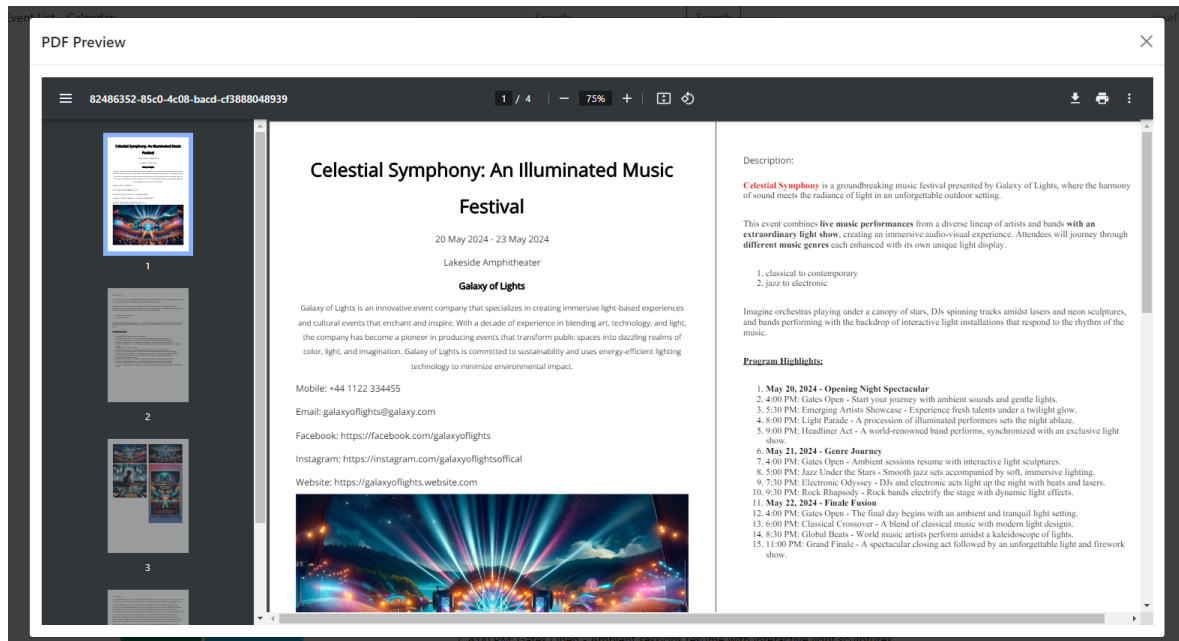
Own Message

Preview PDF

Obrázek 25 Modální okno pro vytvoření závěrečné zprávy

Po kliknutí na „Preview PDF“ se zobrazí standardní zobrazení PDF závěrečné zprávy, kde si uživatel může zkontrolovat, jak závěrečná zpráva vypadá.

Závěrečnou zprávu si tak následně může vytisknout, uložit nebo se případně vrátit zpět a upravit, jaké údaje budou viditelné v závěrečné zprávě.



Obrázek 26 Návrh závěrečné zprávy

Celá vygenerovaná závěrečná zpráva je k nahlédnutí v příloze (Příloha P II).

PDF je generované pomocí knihovny iText 7. V kódu tak nejdříve dojde k vygenerování objektu Document a následně se vytvoří specifický font, který se v dokumentu využívá.

Poté se do dokumentu dynamicky přidávají texty a obrázky podle toho, jaké pole si uživatel definoval pro zobrazení. U obrázků se navíc upravuje velikost a jejich zobrazení.

Nakonec je dokument uzavřen a výsledek je vrácen jako pole bajtů.

```
public async Task<byte[]> GenerateEventReport1Async(EventDetailViewModel model, PdfReportPreferences preferences)
{
    using (var memoryStream = new MemoryStream())
    {
        using (var writer = new PdfWriter(memoryStream))
        {
            using (var pdf = new PdfDocument(writer))
            {
                var document = new Document(pdf);
                Style normal = new Style();
                var fontPath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "fonts", "OpenSans-Regular.ttf");
                PdfFont font = PdfFontFactory.CreateFont(fontPath, PdfEncodings.IDENTITY_H, PdfFontFactory.EmbeddingStrategy.PREFER_EMBEDDED);
                bool isDevelopment = _env.IsDevelopment();
                // First Page
                if (preferences.IncludeTitle)
                {
                    document.Add(new Paragraph($"{model.EventName}").SetFont(font).SetFontSize(25).SetBold().SetTextAlignment(TextAlignment.CENTER));
                }

                if (preferences.IncludeCompanyDetail)
                {
                    var company = await _context.Companies.FindAsync(model.CompanyId.Value);
                    if (company != null)
                    {
                        document.Add(new Paragraph($"{company.Name}").SetFont(font).SetTextAlignment(TextAlignment.CENTER).SetBold());
                        document.Add(new Paragraph($"{company.Description}").SetFont(font).SetTextAlignment(TextAlignment.CENTER).SetFontSize(10));
                    }
                }

                if (preferences.IncludeContact)
                {
                    if (!string.IsNullOrEmpty(model.Facebook))
                    {
                        document.Add(new Paragraph($"Facebook: {model.Facebook}").SetFont(font));
                    }
                    if (!string.IsNullOrEmpty(model.Instagram))
                    {
                        document.Add(new Paragraph($"Instagram: {model.Instagram}").SetFont(font));
                    }
                }
            }
        }
    }
}
```

Obrázek 27 Ukázka části kódu implementace generování závěrečné zprávy

7.6 Nastavení společnosti

Nastavení společnosti umožňuje uživatelům, konkrétně členům společnosti, spravovat její informace a členy. Tato část je přístupná přes dropdown menu, které se otevírá kliknutím na uživatelskou e-mailovou adresu v hlavní navigaci.

Stránka nastavení společnosti je organizovaná do několika záložek, které jsou přístupné prostřednictvím vlastní navigace umístěné na stránce:

- Overview (Přehled): Zobrazuje základní informace o společnosti, mezi které patří název a popis společnosti. Tuto záložku mohou vidět všichni členové, ale pouze administrátoři mají možnost informace editovat.
- Users (Uživatelé): Tato záložka je určena pro správu uživatelů společnosti a jejich rolí. Administrátoři zde mohou přidávat a odebírat uživatele a nastavovat jejich role v rámci společnosti.
- Contact (Kontaktní údaje): Zobrazuje kontaktní údaje společnosti, mezi které patří telefonní číslo, e-mailová adresa a odkazy na sociální sítě. Tuto záložku mohou vidět a editovat pouze administrátoři a manažeři.

Navigace mezi těmito oblastmi je realizována pomocí Bootstrap navigační lišty. Každá záložka v navigaci je spojena s odpovídající částí stránky, která se dynamicky zobrazuje nebo skrývá podle toho, která záložka je aktivní.

Tento efekt je dosažen pomocí CSS vlastnosti `display:none` pro skryté sekce.

```
<nav class="navbar navbar-expand-sm navbar-light bg-light" style="border-bottom: 1px solid #gray;border-top: 1px solid #gray;">
  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#companyNavbar"
    aria-controls="companyNavbar" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="companyNavbar">
    <a class="nav-link @(view == "overview" ? "active" : "")" href="?view=overview">Overview</a>
    <a class="nav-link @(view == "users" ? "active" : "")" href="?view=users">Users</a>
    <a class="nav-link @(view == "contact" ? "active" : "")" href="?view=contact">Contact</a>
  </div>
</nav>
<br />
<div id="overview-section" style="@{(view == "overview" ? "" : "display:none;")}">
  <h2>@Model.Company.Name</h2>
  <p>@Model.Company.Description</p>
</div>

<div id="users-section" style="@{(view == "users" ? "" : "display:none;")}">
  ...
</div>

<div id="contact-section" style="@{(view == "contact" ? "" : "display:none;")}">
  ...
</div>
```

Obrázek 28 Ukázka přepínání mezi jednotlivými záložkami v detailu společnosti

Jednou z funkcionalit pro administrátory je možnost přidat nové uživatele do společnosti přes tlačítko „Add User“ v záložce Users. Po kliknutí na tlačítko se otevře modální okno s formulářem pro zadání e-mailové adresy.

Po zadání e-mailové adresy a potvrzení se spustí javaskriptová funkce `addUserToCompany`. Ta odešle dotaz do kontroleru `CompanyController`, kde se provádí validace e-mailové adresy na straně serveru. Nejprve je kontrolováno, zda existuje e-mailová adresa v tabulce registrovaných uživatelů v aplikaci a následně se kontroluje, zda uživatel není již členem dané společnosti.

```

function addUserToCompany() {
    var email = document.getElementById("userEmail").value;
    var companyId = '@Model.Company.Id';

    fetch(`/Company/AddUserToCompany?email=${encodeURIComponent(email)}&companyId=${companyId}`, {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        }
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            notification.show('success', 'User added successfully.', true);
        } else {
            alert("Failed to add user: " + data.message);
        }
    })
    .catch(error => {
        console.error('Error:', error);
        alert("An error occurred. Please try again.");
    });
}

```

Obrázek 29 Javascript funkce pro přidání člena do společnosti

V případě, že obě validace jsou v pořádku, tak se vytvoří nový záznam do tabulky UserCompanies, kde se uloží ID uživatele, ID společnosti, do které byl přidáný, a role Member, která se považuje za výchozí a následně se vrátí stav do skriptu.

```

[HttpPost]
Počet odkazů: 0
public async Task<IActionResult> AddUserToCompany(string email, int companyId)
{
    var user = await _userManager.FindByEmailAsync(email);
    if (user == null)
    {
        return Json(new { success = false, message = "User not found." });
    }

    var existingMember = await _context.UserCompanies
        .AnyAsync(uc => uc.UserId == user.Id && uc.CompanyId == companyId);
    if (existingMember)
    {
        return Json(new { success = false, message = "User is already a member of the company." });
    }

    var userCompany = new UserCompany
    {
        UserId = user.Id,
        CompanyId = companyId,
        Role = "Member"
    };

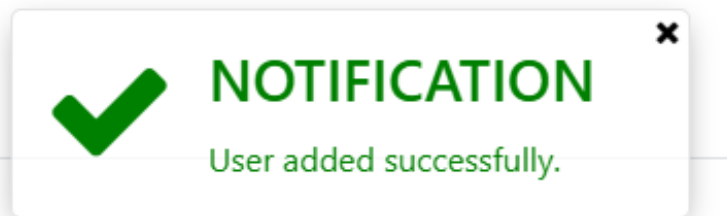
    _context.UserCompanies.Add(userCompany);
    await _context.SaveChangesAsync();

    return Json(new { success = true });
}

```

Obrázek 30 Implementace metody AddUserToCompany

V případě, že výsledek byl úspěšný, tak se zobrazí notifikační zpráva, která informuje uživatele o tom, že činnost, kterou provedl, byla úspěšná. Notifikace je vytvořena dynamicky a zobrazuje se v levém spodním rohu obrazovky. Pokud uživatel notifikaci neuzavře manuálně, tak se uzavře automaticky po 5 sekundách.



Obrázek 31 Ukázka zobrazené notifikační zprávy

```

var floatingMessage = {
  messageTemplate: $('<div class="floating-message"></div>'),
  init: function () {
    if ($('#floating-message-container').length === 0) {
      $('body').append('<div id="floating-message-container"></div>');
    }
  },
  showMessage: function (type, title, text) {
    this.init();

    let msg = this.messageTemplate.clone().show();
    msg.addClass(type);
    msg.html('<i class="fa ${type === 'success' ? 'fa-check' : 'fa-times'}"></i> <div class="msg-body">
    <h3>${title}</h3><p>${text}</p></div><a class="dismiss-button fa fa-times"></a>');

    $('#floating-message-container').append(msg);
    msg.fadeIn();

    setTimeout(function () {
      msg.fadeOut(function () {
        $(this).remove();
      });
    }, 5000);

    msg.find('.dismiss-button').on('click', function () {
      $(this).parent().fadeOut(function () {
        $(this).remove();
      });
    });
  }
};

var notification = {
  show: function (type, message, shouldReload = false) {
    if (shouldReload) {
      sessionStorage.setItem('notificationType', type);
      sessionStorage.setItem('notificationMessage', message);
      location.reload();
    } else {
      floatingMessage.showMessage(type, 'Notification', message);
    }
  }
};

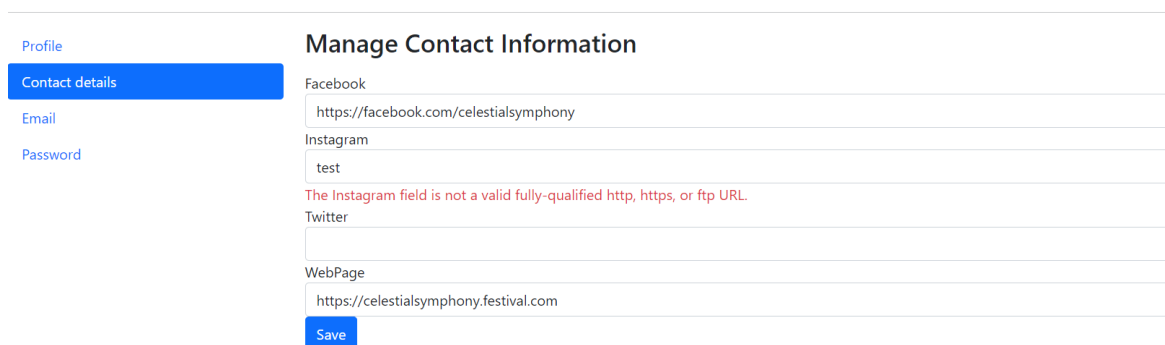
```

Obrázek 32 Implementace notifikační zprávy

7.7 Nastavení uživatele

Uživatelské nastavení slouží k nastavení, popřípadě změně osobních a přihlašovacích údajů uživatele. Samotné nastavení je z podstatné části vygenerované pomocí ASP.NET Core Identity. Pro potřeby aplikace byla přidána záložka Contact details, která slouží pro upravení URL adres na sociální sítě uživatele. Záložka Profile slouží ke změně telefonního čísla. Záložka Email slouží k možnosti změny e-mailové adresy a záložka Password umožňuje uživateli změnit si své heslo do webové aplikace.

Manage your account



The screenshot shows a user account management interface. On the left, a sidebar contains links for 'Profile', 'Contact details' (highlighted in blue), 'Email', and 'Password'. The main content area is titled 'Manage Contact Information' and contains four input fields: 'Facebook' with the value 'https://facebook.com/celestialsymphony', 'Instagram' with the value 'test', 'Twitter' (empty), and 'WebPage' with the value 'https://celestialsymphony.festival.com'. A red error message is displayed below the Instagram field: 'The Instagram field is not a valid fully-qualified http, https, or ftp URL.' A blue 'Save' button is located at the bottom of the form.

Obrázek 33 Nastavení uživatele

7.8 Seznam akcí

V seznamu akcí uživatel vidí všechny akce, kterými je organizátor anebo se nachází ve společnosti, která je organizátorem akce. Samotná stránka obsahuje tlačítko pro vytváření akcí, možnost vyhledávání akcí podle názvu a filtrování akcí od nejstarší/nejnovější.

Kulturní akce obsahují podobné údaje jako na hlavní stránce, a to hlavní obrázek akce, název akce, popis, datum a místo konání. Zároveň se zde u každé akce nachází tlačítka pro editování akce, zobrazení statistik a smazání akce. U akcí, kde je organizátorem společnost a uživatel nemá vyšší právo než Member, se mu při prokliknutí přes tlačítko „Stats“ a „Edit“ zobrazí, že k těmto činnostem nemá přístup.

Access denied

You do not have access to this resource.

Obrázek 34 Ukázka obrazovky při nedostatečných právech uživatele

Tato činnost je prováděna vždy na začátku metody pomocí if podmínky, která volá metodu `UserHasPermission`, kde dochází ke kontrole práv uživatele podle role ve společnosti nebo podle toho, zda je on sám organizátor akce.

```
public async Task<IActionResult> EventStats(int id)
{
    if (!await UserHasPermission(id, new[] { "Admin" }, true))
    {
        return Forbid();
    }

    // pokračování kódu
}

Počet odkazů: 3
private async Task<bool> UserHasPermission(int eventId, string[] allowedRoles = null, bool mustBeOrganizer = false)
{
    var userId = _userManager.GetUserId(User);
    var cultureEvent = await _context.CultureEvents.FindAsync(eventId);

    if (cultureEvent == null)
        return false; // Event does not exist

    if (mustBeOrganizer && cultureEvent.UserId == userId)
        return true; // User is the organizer of the event

    if (allowedRoles != null && cultureEvent.CompanyId.HasValue)
    {
        // Check if the user has any of the allowed roles within the company that owns the event
        var userInCompany = await _context.UserCompanies
            .Where(uc => uc.UserId == userId && uc.CompanyId == cultureEvent.CompanyId)
            .Select(uc => uc.Role)
            .AnyAsync(role => allowedRoles.Contains(role));

        return userInCompany;
    }

    return false;
}
```

Obrázek 35 Implementace kontroly práv uživatele

7.8.1 Formulář pro přidání akce

Pomocí formuláře může akci vytvořit jakýkoli uživatel, který je do systému přihlášený. V rámci formuláře může uživatel definovat přes dropdown „Event Creator“, kdo je skutečným organizátorem akce. Může zvolit možnost Personal, která znamená, že bude u akce on sám uveden jako organizátor nebo může vybrat ze seznamu společnost, ve které se nachází a má právo v rámci ní akci vytvořit. Formulář dále obsahuje pole pro specifikaci detailů akce, jako název akce, místo a čas konání a další.

Pro popis akce je použit WYSIWYG editor Quill.js, který umožňuje aplikovat různé formátování textu a docílit tak možnosti lépe strukturovat popis a zvýraznit tak důležité informace týkající se akce.

Editor je implementován prostřednictvím jednoduché inicializace na prázdném div elementu s unikátním Id. Společně s ním je definovaný toolbar, kde je rozvrženo, které možnosti formátování a v jakém pořadí se mají zobrazovat.

Při vytvoření nové akce se popis akce uloží ve formátu HTML kódu, který se pak následně v jiných částech aplikace zobrazuje s formátováním anebo se toto formátování přes jednoduchou funkci odstraní.

```
var toolbarOptions = [
  [ { 'font': [] } ],
  [ { 'size': [ 'small', false, 'large', 'huge' ] } ],
  [ 'bold', 'italic', 'underline' ],
  [ { 'list': 'ordered' }, { 'list': 'bullet' } ],
  [ { 'color': [] }, { 'background': [] } ],
  [ { 'align': [] } ],
  [ 'clean' ]
];

const quill = new Quill('#editor', {
  theme: 'snow',
  modules: {
    toolbar: toolbarOptions
  }
});

function handleSubmit() {
  var html = quill.root.innerHTML;

  document.getElementById('description').value = html;

  return true;
}
```

Obrázek 36 Implementace WYSIWYG editoru s toolbarem

Nakonec může organizátor k akci přidat libovolný počet příloh, které musí být ve formátu obrázku nebo videa společně s jedním obrázkem, který je definovaný jako hlavní. Ten se pak zobrazuje na hlavní stránce.

Event Creator:

Event Name:

Date:
From:
To:

Place:

Tickets:
Ticket Name Price

Description:

Important Description:

Attachments:

Main picture:

Obrázek 37 Formulář pro vytvoření nové akce

7.8.1.1 Ukládání souborů v akci

Na serverové části aplikace je definovaná metoda `SaveFileAsync`. Cílem této metody je rozlišit, zda se uživatel nachází v produkční nebo vývojové verzi aplikace a podle toho zvolit přístup, jakým bude soubor uložený. V případě, že se nachází na vývojové verzi, tak se soubor uloží do složky `Uploads`, která se nachází ve složce `wwwroot` v projektu aplikace.

```
private async Task<string> SaveFileAsync(IFormFile file)
{
    if (_env.IsProduction())
    {
        var blobName = Guid.NewGuid().ToString() + Path.GetExtension(file.FileName);
        using (var stream = file.OpenReadStream())
        {
            var blobUri = await _blobService.UploadFileAsync(stream, blobName);
            return blobUri.ToString();
        }
    }
    else
    {
        var uploadsDirectoryPath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "Uploads");
        if (!Directory.Exists(uploadsDirectoryPath))
        {
            Directory.CreateDirectory(uploadsDirectoryPath);
        }
        var fileName = Guid.NewGuid().ToString() + Path.GetExtension(file.FileName);
        var filePath = Path.Combine(uploadsDirectoryPath, fileName);

        using (var fileStream = new FileStream(filePath, FileMode.Create))
        {
            await file.CopyToAsync(fileStream);
        }

        return Path.Combine("Uploads", fileName);
    }
}
```

Obrázek 38 Implementace metody `SaveFileAsync`

Pokud je uživatel na produkční verzi aplikace, tak se pro uložení využije cloudové úložiště Azure. V metodě `SaveFileAsync` se zavolá metoda `UploadFileAsync` ve službě `BlobService`, která slouží pro uložení souboru přímo do kontejneru v úložišti. V souboru `appsettings.json` je konfigurace `AzureStorageConfig`, kde je definovaný `ConnectionString` k úložišti společně s názvem kontejneru, názvem úložiště, a odkazu na úložiště.

```
public class BlobService : IBlobService
{
    private readonly string _connectionString;
    private readonly string _containerName;

    Počet odkazů: 0
    public BlobService(IOptions<AzureBlobStorageOptions> options)
    {
        _connectionString = options.Value.ConnectionString;
        _containerName = options.Value.ContainerName;
    }

    Počet odkazů: 2
    public async Task<string> UploadFileAsync(Stream fileStream, string fileName)
    {
        var uniqueFileName = GenerateUniqueFileName(fileName);
        var blobServiceClient = new BlobServiceClient(_connectionString);
        var blobContainer = blobServiceClient.GetBlobContainerClient(_containerName);
        await blobContainer.CreateIfNotExistsAsync();
        var blobClient = blobContainer.GetBlobClient(uniqueFileName);
        await blobClient.UploadAsync(fileStream, true);
        return blobClient.Uri.ToString();
    }

    Počet odkazů: 1
    private string GenerateUniqueFileName(string originalFileName)
    {
        var extension = Path.GetExtension(originalFileName);
        var fileNameWithoutExtension = Path.GetFileNameWithoutExtension(originalFileName);
        var uniqueFileName = $"{fileNameWithoutExtension}_{Guid.NewGuid()} {extension}";
        return uniqueFileName;
    }
}
```

Obrázek 39 Implementace služby BlobService

Podobným způsobem je následně řešený proces mazání souborů, kdy se opět kontroluje, v jakém prostředí uživatel je, a podle toho se určí odkud se obrázek smaže.

7.8.2 Formulář pro editaci akce

Formulář pro editaci akce je navržen podobně jako formulář pro vytvoření akce, ale obsahuje několik specifických úprav. Na rozdíl od formuláře pro vytvoření akce, kde jsou všechna data zadávána nově, formulář pro editaci načítá a zobrazuje stávající informace o akci, které lze upravit.

Jedním z hlavních rozdílů je způsob, jakým formulář zpracovává přílohy. Ve formuláři se stejně jako při vytváření akce zobrazuje tlačítko pro přidání nových příloh, ale současně s ním se zobrazuje seznam všech příloh, které jsou již uloženy k dané akci.

Každá příloha disponuje zaškrtačím boxem, pomocí kterého může uživatel soubor odstranit. V případě, že by chtěl organizátor změnit úvodní obrázek, tak stačí nahrát nový obrázek a při uložení se automaticky smaže předchozí hlavní obrázek, pokud existoval.

```
private async Task DeleteFileAsync(string filename, string? localfile)
{
    if (!_env.IsProduction())
    {
        await _blobService.DeleteFileAsync(filename);
    }
    else
    {
        var uploadsDirectoryPath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "Uploads");
        var filePathToDelete = Path.Combine(uploadsDirectoryPath, localfile.Replace("Uploads\\", ""));

        if (System.IO.File.Exists(filePathToDelete))
        {
            System.IO.File.Delete(filePathToDelete);
        }
    }
}
```

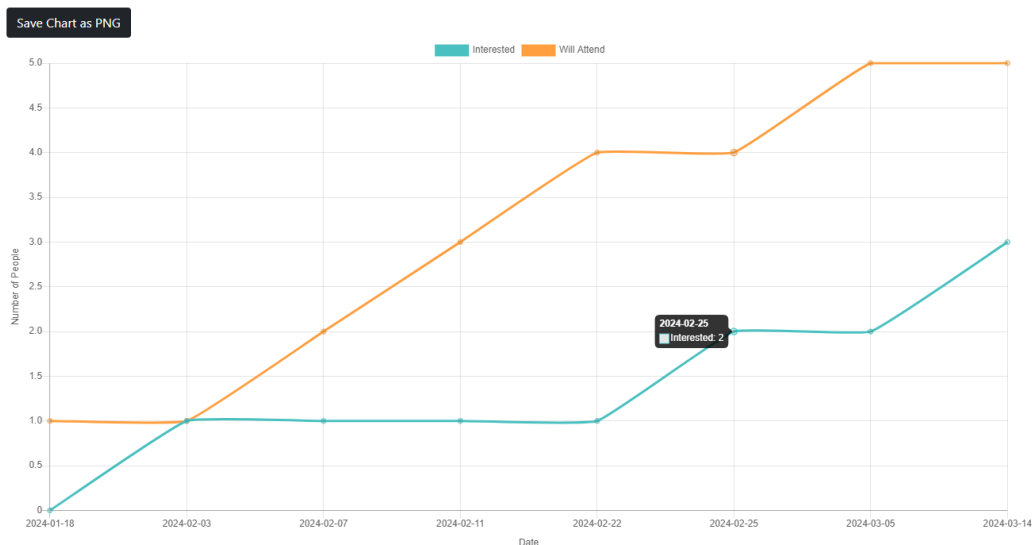
Obrázek 40 Implementace metody pro mazání souborů

7.8.3 Statistika akce

Organizátoři akce mají možnost zobrazit statistiku zájmu o konkrétní akci v podobě grafu. Tento graf je generován pomocí JavaScriptové knihovny Chart.js, kde osa X reprezentuje čas (datумы) a osa Y ukazuje celkový počet uživatelů, kteří projeví zájem o akci.

V grafu jsou zobrazeny dvě lineární křivky. Modrá křivka znázorňuje počet uživatelů, kteří o danou akci projeví zájem (Interested) a oranžová křivka ukazuje počet uživatelů, kteří se akce chystají zúčastnit (Will Attend).

Cílem této funkcionality je přinést organizátorům nástroj, jak sledovat vývoj zájmu o akci v čase a identifikovat tak, jaké události měly na zájem vliv. Organizátor tak získává nástroj, na základě kterého může posoudit, zda případná propagace měla na akci pozitivní nebo negativní vliv. Zároveň organizátoři mohou předběžně očekávat, jak velká bude účast na akci.



Obrázek 41 Graf zájmu uživatelů

Implementace zobrazení statistik začíná na serverové straně, kdy se první spustí metoda `EventStats` v kontroleru `EventsController`. Tato metoda nejprve získá data o uživatelích, kteří projeví zájem o danou akci.

Konkrétně data rozdělí do dvou skupin, kde v první skupině jsou data o uživatelích, kteří pouze projeví zájem o akci a druhou skupinu uživatelů, kteří se akce mají v plánu zúčastnit. Data jsou následně zpracována tak, aby v grafu zobrazovala kumulativní růst nebo pokles počtu zájemců v čase místo zobrazení změn v počtu zájmu v konkrétní dny.

```
public async Task<IActionResult> EventStats(int id)
{
    var eventData = await _context.EventUsers
        .Where(eu => eu.EventId == id)
        .GroupBy(eu => eu.StatusDateTime.Date)
        .Select(group => new
        {
            Date = group.Key,
            InterestedCount = group.Count(eu => eu.Status == "Interested"),
            WillAttendCount = group.Count(eu => eu.Status == "WillAttend")
        })
        .OrderBy(x => x.Date)
        .ToListAsync();

    var cumulativeInterestedCounts = new List<int>();
    var cumulativeWillAttendCounts = new List<int>();
    int runningTotalInterested = 0;
    int runningTotalWillAttend = 0;
    foreach (var day in eventData)
    {
        runningTotalInterested += day.InterestedCount;
        cumulativeInterestedCounts.Add(runningTotalInterested);

        runningTotalWillAttend += day.WillAttendCount;
        cumulativeWillAttendCounts.Add(runningTotalWillAttend);
    }

    var model = new EventStatsViewModel
    {
        Dates = eventData.Select(x => x.Date).ToList(),
        InterestedCounts = cumulativeInterestedCounts,
        WillAttendCounts = cumulativeWillAttendCounts
    };

    return View(model);
}
```

Obrázek 42 Implementace kódu pro statistiky akce

Na straně klienta se používá javascript spolu s open-source knihovnou `Chart.js` pro vykreslení grafu. V rámci inicializace grafu se zvolí typ grafu na lineární (`line`) a následně se vytvoří dva datasety (křivky) s daty získanými z kontroleru, a nakonec se nastaví názvy osy X a osy Y.

```
<script src="https://cdn.jsdelivr.net/npm/chart.js@2.9.4/dist/chart.js"></script>
<script>
  const ctx = document.getElementById('myChart').getContext('2d');
  var eventStatsChart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: @Html.Raw(Json.Serialize(Model.Dates.Select(date => date.ToString("yyyy-MM-dd")))),
      datasets: [{
        label: 'Interested',
        data: @Html.Raw(Json.Serialize(Model.InterestedCounts)),
        borderColor: 'rgb(75, 192, 192)',
        tension: 0.1,
        fill: false
      },
      {
        label: 'Will Attend',
        data: @Html.Raw(Json.Serialize(Model.WillAttendCounts)),
        borderColor: 'rgb(255, 159, 64)',
        tension: 0.1,
        fill: false
      }
    ]
  }
},
options: {
  scales: {
    yAxes: [{
      scaleLabel: {
        display: true,
        labelString: 'Number of People'
      }
    }],
    xAxes: [{
      scaleLabel: {
        display: true,
        labelString: 'Date'
      }
    }
  ]
}
});
</script>
```

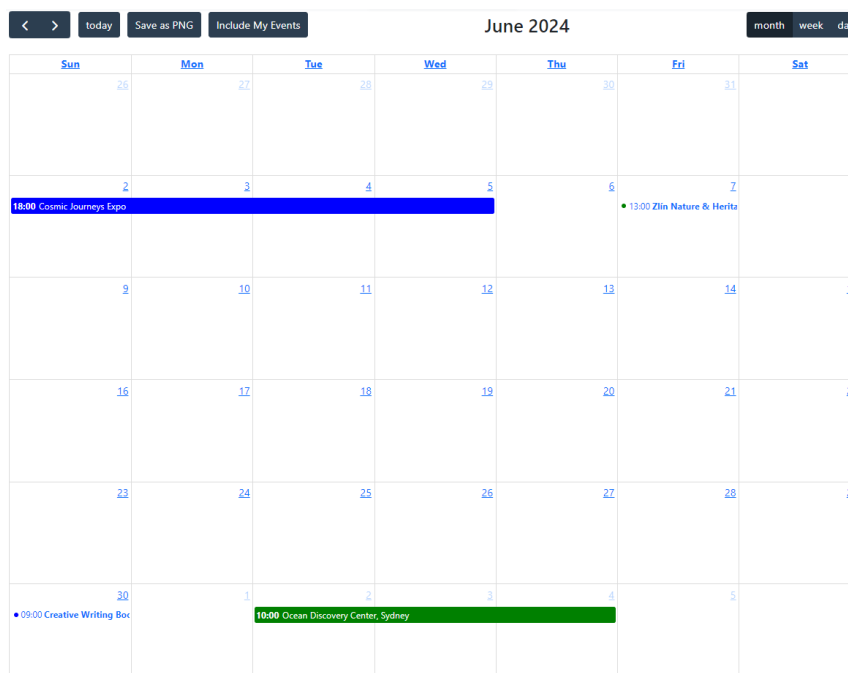
Obrázek 43 Vizualizace účasti na akci pomocí Chart.js

7.9 Kalendář akcí

Kalendář akcí je dostupný pro přihlášené uživatele z hlavní navigace stránky. Samotný kalendář je vykreslený pomocí knihovny FullCalendar. Tento kalendář je rozšířený o tlačítka „Save as PNG“ pro vytvoření obrázku kalendáře a „Include My Events“.

Ve výchozím zobrazení se v kalendáři zobrazují veškeré akce, u kterých uživatel potvrdil, že se jich zúčastní anebo o ně má zájem. Akce, u kterých má uživatel potvrzenou účast, jsou v kalendáři označeny modře, zatímco ty, o které má zájem, jsou označeny zeleně.

Při zakliknutí možnosti „Include My Events“ se do kalendáře přidají i všechny akce, u kterých je uživatel organizátorem anebo je ve společnosti, která je organizátorem dané akce. Tyto akce se v kalendáři vyznačují červenou barvou.



Obrázek 44 Zobrazení kalendáře s akcemi

Kalendář je inicializován v souboru skriptů, který je začleněn na konci HTML stránky pod sekci `@section scripts`. Zde je vytvořen objekt kalendáře, který je svázán s HTML elementem s ID `calendar`.

Konfigurace zahrnuje nastavení výchozího zobrazení (`dayGridMonth`), počátečního data a nástrojové lišty, která obsahuje tlačítka pro navigaci a vlastní tlačítka pro export kalendáře do PNG a zahrnutí vlastních událostí uživatele.

Události jsou načítány dynamicky ze serveru pomocí AJAX volání, které je specifikováno v konfiguraci kalendáře. URL pro načítání událostí je generováno pomocí `@Url.Action()`.

Přepínání kalendáře mezi zobrazením událostí, u kterých uživatel projevil zájem a zobrazením, kde se navíc zobrazují události, kde je organizátorem se řídí pomocí přepínače, který mění stav proměnné `includeMyEvents`. Podle stavu proměnné se následně zvolí zdroj událostí, které se v kalendáři zobrazují.

Na serverové straně, v `CalendarController`, jsou definovány metody pro získání událostí. `GetEvents` a `GetMyEvents` jsou dvě klíčové metody, které filtrují události na základě role a zájmu uživatele. Tyto metody využívají Entity Framework pro dotazování do databáze a následně vrací data ve formátu, který je kompatibilní s FullCalendar (např. název události, datum začátku a konce, URL detailu události a barva reprezentující stav uživatele vůči události).

8 TESTOVÁNÍ V APLIKACI CULTUR4U

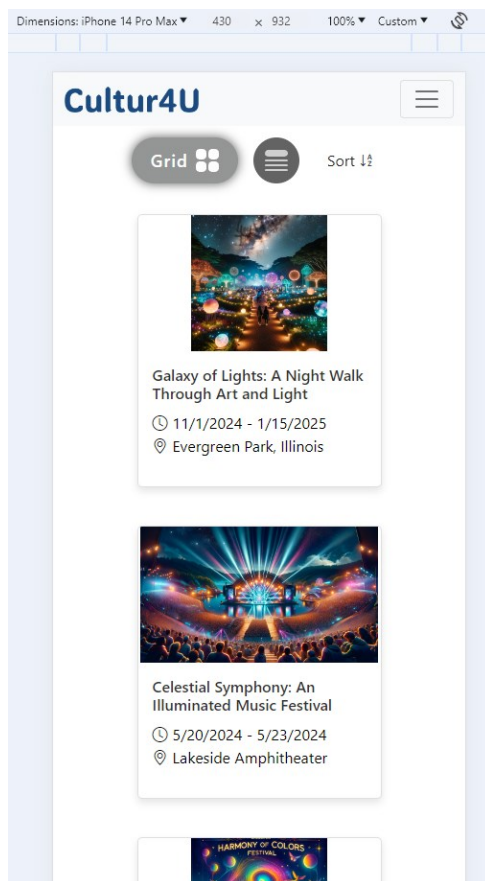
V rámci vývoje aplikace Cultur4U bylo primárně použito manuální testování. Tento přístup byl zvolen kvůli jeho flexibilitě a schopnosti rychle identifikovat chyby jak ve funkčnosti, tak v estetice.

Manuální testování probíhalo průběžně během celého procesu vývoje, což umožnilo okamžité opravy nalezených chyb. Testování probíhalo jak ve vývojové, tak i v produkční verzi nasazené na Azure, aby se ověřilo, že nedošlo k nežádoucím změnám během publikace, a že se aplikace chová konzistentně v obou prostředích.

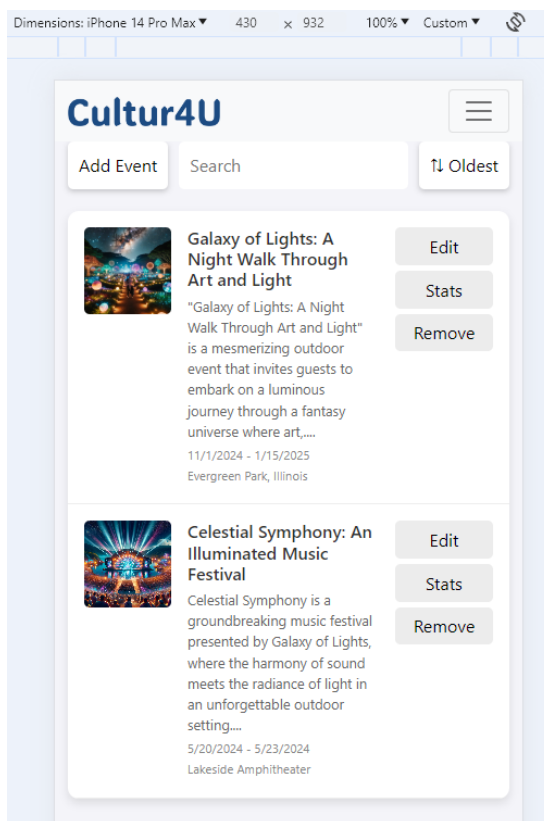
Důležitým aspektem testování bylo ověření funkčnosti ukládání a zobrazování obrázků a dalších souborů, jelikož tato činnost je odlišně prováděna ve vývojové verzi, kde se soubory ukládají přímo do složky s projektem a v produkční verzi, kde se soubory ukládají do Azure Blob Storage.

Dalším způsobem testování aplikace bylo testování samotného zobrazení uživatelského rozhraní. Primárně byl kladen důraz na to, aby se všechny prvky na stránce zobrazovaly tak, jak mají a nedocházelo k nežádoucím jevům jako je například přetékaní textu, chybné zalomení textu a špatné umístění prvků na stránce. K tomuto testování byl využit nástroj zabudovaný přímo ve vývojářských nástrojích v Google Chrome.

Dále došlo k ověření, zda stránka správně vypadá a funguje i v jiných webových prohlížečích. Otestováno bylo správné fungování v prohlížečích Safari, Mozilla Firefox, Microsoft Edge a Google Chrome.



Obrázek 45 Mobilní zobrazení domovské stránky



Obrázek 46 Mobilní zobrazení stránky EventList

Posledním způsobem testování bylo vytvoření několika automatických testů pro kritické části aplikace. V rámci vývoje aplikace Cultur4U byl vytvořen nový projekt, který využil šablonu „Projekt testů xUNIT“ přímo z Visual Studio 2022.

V tomto projektu bylo vytvořeno několik souborů, které seskupují testy pro konkrétní oblasti. Jedním z příkladů je soubor EventsController.cs. Tento soubor obsahuje jednotkové testy pro otestování činností souvisejících s kulturními akcemi, jako jsou vytvoření a editování akce nebo zobrazení seznamu a detailu akce.

Testy jsou realizované pomocí knihovny Moq s využitím in-memory databáze pro simulaci databázových operací bez nutnosti připojení k fyzické databázi.

Jedním z testů je například test CreateEvent_Post_ValidData_CreatesEventAndRedirects, který kontroluje, že při poslání validních dat metodou POST dojde k úspěšnému vytvoření akce a následnému přesměrování na stránku EventList.

Před samotným testem se provede inicializace s nastavením mock závislostí a připravením in-memory databáze. Následně se testuje výsledek zavolání metoda CreateEvent s definovanými vstupními hodnotami. Poté se pomocí Assert ověřuje výsledek z metody a zda uživatel byl přesměrován na správnou stránku po vytvoření akce.

```
Počet odkazů: 5 | Úspěšně: 5/5
private void SetupController()
{
    _context = CreateNewContext();
    _controller = new EventsController(_mockLogger.Object, _mockUserManager.Object,
        _context, _mockBlobService.Object, _mockWebHostEnvironment.Object);
}
Počet odkazů: 1
private ApplicationDbContext CreateNewContext()
{
    var options = new DbContextOptionsBuilder<ApplicationDbContext>()
        .UseInMemoryDatabase(Guid.NewGuid().ToString())
        .Options;
    return new ApplicationDbContext(options);
}

[Fact]
Počet odkazů: 0
public async Task CreateEvent_Post_ValidData_CreatesEventAndRedirects()
{
    SetupController();

    var result = await _controller.CreateEvent(
        "New Event", DateTime.Today, DateTime.Today.AddDays(1),
        TimeSpan.FromHours(9), TimeSpan.FromHours(17), "Venue",
        "Description", "Important Info", null, null,
        new[] { "General Admission" }, new[] { 10.00M }, "Personal") as RedirectToActionResult;

    Assert.NotNull(result);
    Assert.Equal("EventList", result.ActionName);
}
```

Obrázek 47 Ukázka implementace jednotkového testu

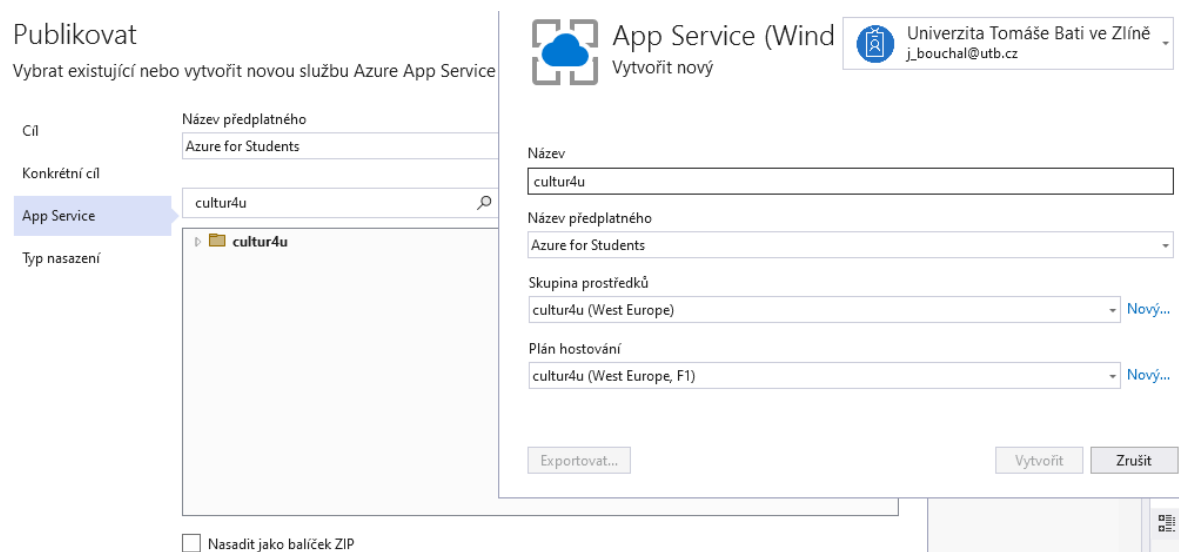
9 PUBLIKACE APLIKACE DO MICROSOFT AZURE

Pro publikaci webové aplikace byla zvolena platforma Microsoft Azure. Primárním důvodem zvolení této platformy je integrace s vývojovým prostředím Microsoft Visual Studio 2022, které poskytuje podporu publikace na Azure, což podstatně zjednodušuje celý proces nasazování webové aplikace.

Prvním krokem bylo otevření kontextového menu pomocí kliknutí pravého tlačítka na název projektu v průzkumníku řešení. Z tohoto menu byla následně zvolena možnost „Publikovat“.

V průvodci publikací byl jako cíl vybrán „Azure App Service (Windows)“. Následně byl vytvořen nový App Service na bezplatném/studentském účtu Azure.

Po jeho vytvoření byl tento App Service specifikován jako cíl publikace a byla zvolena možnost „Publikovat (vygeneruje soubor pubxml)“, čímž byl proces vytvoření profilu publikace dokončen.



Obrázek 48 Modální okno pro vytvoření nové App Service

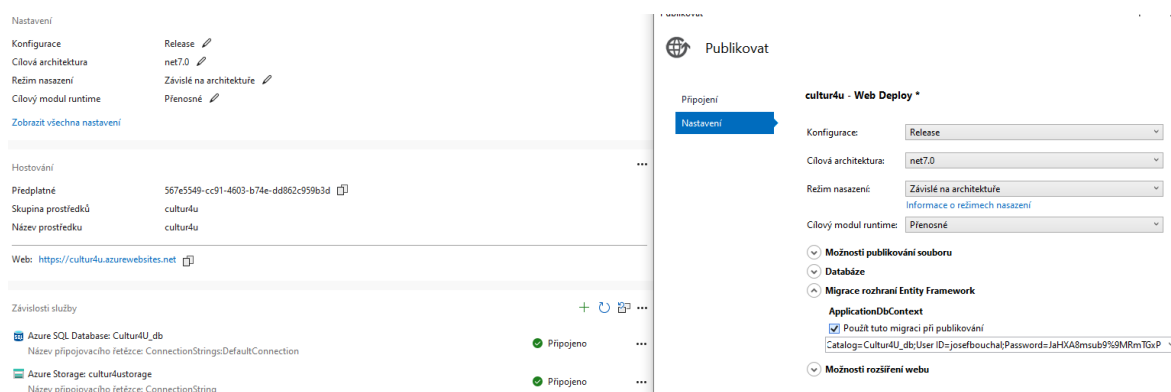
Dalším krokem po vytvoření profilu bylo začlenění databáze pro ukládání dat a úložiště pro ukládání médií. Opět podobně jako v rámci samotného webu byla zvolena databázová služba Azure SQL Database.

Při vytváření nové databáze je potřeba nejprve vytvořit nový databázový server, kde je potřeba si definovat přístupové údaje pro správce. Následně stačí nakonfigurovat samotnou databázi, kde je opět potřeba vytvořit si přístupové údaje a definovat připojovací řetězec, kterým se propojuje databáze se samotnou aplikací.

Pro ukládání souborů bylo podobným způsobem vytvořeno Azure úložiště, které se připojilo k závislostem služby. Po úspěšném připojení bylo potřeba si otevřít konfigurační rozhraní úložiště přes portal azure.

V tomto rozhraní bylo potřeba povolit anonymní přístup k blob souborům a poté vytvořit nový kontejner s nastavením úrovně soukromí na Blob. Samotné kontejnery umožňují lépe organizovat umístování konkrétních souborů a změna soukromí je potřebná pro to, aby uživatel měl možnost soubory nejen ukládat, ale i mazat a zobrazovat.

Posledním krokem před spuštěním publikace je otevření okna všech nastavení profilu publikace, kde je potřeba povolit použití migrací při publikování. Díky tomuto po publikaci webové aplikace dojde i k inicializaci databáze a všech tabulek, které máme definované. A proto není potřeba strukturu databáze vytvářet manuálně.



Obrázek 49 Možnost povolení migrace při publikaci

Nakonec už stačí pouze doplnit nastavení AzureStorageConfig do souboru appsettings.json a poté spustit pomocí vytvořeného profilu publikace samotné nasazení webové aplikace. Webová aplikace je dostupná na URL adrese <https://cultur4u.azurewebsites.net/> po dobu obhajob bakalářských prací.

10 ZHODNOCENÍ A MOŽNÁ VYLEPŠENÍ

Tato kapitola je zaměřena na celkové zhodnocení webové aplikace a také jsou zde navržena další možná vylepšení.

10.1 Zhodnocení výsledků

Praktická část práce popsala postup vývoje aplikace Cultur4U od analýzy funkčních a nefunkčních požadavků přes návrh a implementaci databáze pomocí přístupu code first v EF až po konkrétní popsání jednotlivých stránek aplikace s popsáním fungování podstatných částí systému. Pro vývoj webové aplikace byla použita architektura ASP.NET Core s MVC strukturou.

Webová aplikace Cultur4U umožňuje uživatelům možnost registrovat se do systému. Uživatelé mohou vytvářet kulturní akce jako jednotlivci anebo mohou vytvořit společnosti s více uživateli a pod touto společností vytvářet akce. Organizátoři akcí mohou své akce libovolně upravovat, mazat a mají možnost vytvořit si souhrnnou závěrečnou zprávu s údaji o akci a vlastním závěrečným textem. Mají také možnost sledovat statistiky zájmu o akci, s kterými mohou pracovat při plánování propagace či je využít pro organizační záležitosti. Běžní uživatelé si mohou prohlížet zveřejněné akce, mohou vyhledávat konkrétní akce a zobrazit si jejich detailní informace. Zároveň systém nabízí možnost pro přihlášené uživatele si zakliknout, zda se konkrétní akce chtějí zúčastnit nebo o ni mají zájem. V kalendáři akcí přihlášení uživatelé vidí akce, o které projevíli zájem, díky tomu je pro ně snadnější hlídat si konkrétní termín konání dané akce.

10.2 Možnosti vylepšení aplikace

Přestože vytvořený informační systém pro evidenci kulturních akcí byl úspěšně nasazen a splňuje základní požadavky a funkcionality, existují oblasti, ve kterých by mohl být systém vylepšen, aby lépe reagoval na potřeby běžných uživatelů, ale i samotných organizátorů.

10.2.1 Lepší filtrování akcí

Systém by mohl být v budoucnu vylepšen o pokročilejší filtrační možnosti, které by uživatelům umožnily snadněji najít akce podle specifických kritérií. Uživatel by mohl filtrovat události dle data, místa konání akce, organizátorů či typu akce.

Zároveň by mohl být implementován systém štítků, kde by organizátor mohl definovat „klíčová slova“, která vystihují jeho akci a následně by běžní uživatelé mohli na základě těchto klíčových slov vyhledávat akce.

10.2.2 Rozšíření statistik

Vedle stávající statistiky pro počet zájemců o konkrétní akci, by bylo užitečné přidat více statistik pro organizátory. Mohlo by se tak jednat o statistiky průměrné účasti na akcích či bodové hodnocení, které by účastníci udělili proběhlé události. To by organizátorům pomohlo budoucí akce přizpůsobit tak, aby lépe vyhovovaly potřebám cílové skupiny.

Nebo například pokud by při registraci uživatele byly mezi povinně vyplňovanými údaji i demografické informace jako věk či pohlaví, mohly by tyto údaje pomoci organizátorům lépe cílit jejich marketingové kampaně pro budoucí akce.

10.2.3 Prodej vstupenek

Ohledně prodeje vstupenek by mohla být zavedena implementace funkcionality pro prodej vstupenek, ať už prostřednictvím vlastního řešení nebo externí služby jako je například Ticketportal. Toto rozšíření by následně umožňovalo další rozšíření statistik.

10.2.4 Rozšíření detailu akce

Dále by mohlo být přidáno více detailů akce. Popis v detailu akce by mohl být rozdělený do jednotlivých sekcí jako jsou program, informace o sponzorech, organizátoři a podobně. Dále by se mohla v detailu zobrazovat interaktivní mapa místa akce. Tato vylepšení by umožnila uživatelům, ale i organizátorům přehlednější zobrazení konkrétní akce.

10.2.5 Sdílení akcí na sociální síť

Následně by webová aplikace mohla být obohacena o sdílení akcí na sociálních sítích pro lepší zviditelnění akce a přilákání širšího publika. Tato funkcionality by měla být navržena tak, že uživatelé umožní sdílet akci s předdefinovanou, případně vlastní zprávou a odkazem na danou akci.

10.2.6 Rozšíření tvorby závěrečné zprávy

Také by systém mohl nabídnout pokročilejší možnosti pro tvorbu závěrečných zpráv včetně možnosti definovat si vlastní rozložení stránek a vizuální styl zprávy. Toto vylepšení by mohlo umožnit organizátorům vytvářet více personalizované dokumenty.

ZÁVĚR

Tato bakalářská práce se zaměřuje na návrh a implementaci informačního systému pro evidenci kulturních akcí, který byl vyvíjen s cílem zjednodušit a zefektivnit správu a propagaci kulturních akcí organizátorům a umožnit běžným uživatelům hledat informace o akcích.

Teoretická část poskytla ucelený přehled o problematice evidování kulturních akcí a představila různá existující řešení na trhu. Následně byly popsány technologie, které byly využity v rámci praktické části. Jsou mezi nimi zahrnuty ASP.NET Core, MVC, Microsoft Visual Studio 2022, SQL, a další, včetně javascriptových knihoven jako Quill.js, Fullcalendar a Chart.js.

Praktická část se soustředila na samotný vývoj webové aplikace nazvané Cultur4U. Byl zdokumentován celý vývojový proces, od analýzy požadavků, přes návrh databáze až po implementaci funkcionalit a uživatelského rozhraní. Aplikace byla otestována jak manuálně, tak pomocí integračních testů, a byla optimalizována pro zobrazení na mobilních zařízeních.

Výsledkem je funkční systém umožňující organizátorům vytváření, editaci a mazání kulturních akcí, jak z pohledu jednotlivců, tak celých společností.

Organizátorům dále poskytuje nástroj pro generování závěrečných zpráv na základě zadaných parametrů, které chtějí ve zprávě zobrazit. Mezi tyto parametry patří zobrazení názvu akce, místa a data konání, kontaktních údajů, údajů o společnosti, fotogalerie a zobrazení hlavního obrázku akce. Dále systém poskytuje nástroj, pomocí kterého si organizátoři mohou zobrazit graf zájmu uživatelů o danou akci, z kterého mohou zjistit, kolik uživatelů se plánuje akce zúčastnit nebo o ni má zájem.

Běžní uživatelé mohou procházet akce, vyhledávat mezi nimi a zobrazovat si detailní informace o nich. Systém jim také umožňuje zobrazit kalendář akcí, o které mají zájem nebo se jich plánují zúčastnit.

V závěru práce jsou navrženy možnosti budoucího rozvoje aplikace, které by rozšířily její funkčnost pro obě cílové skupiny. Tyto kroky by napomohly aplikaci stát se pro uživatele ještě užitečnější a atraktivnější, což by podpořilo růst uživatelské základny.

SEZNAM POUŽITÉ LITERATURY

- [1] HALL, Mark. *Facebook : Overview, History, & Facts*. Online. In: Britannica Money. 2024. Dostupné z: <https://www.britannica.com/money/Facebook>. [cit. 2024-05-05].
- [2] *Facebook*. Online. 2024. Dostupné z: <https://www.facebook.com/>. [cit. 2024-05-05].
- [3] *O Kudy z nudy – Kudy z nudy*. Online. 2024. Dostupné z: <https://www.kudyznudy.cz/o-kudy-z-nudy>. [cit. 2024-05-05].
- [4] *O Pragokoncertu : PRAGOKONCERT BOHEMIA*, a.s.. Online. Dostupné z: <https://www.pragokoncert.com/cs/o-pragokoncertu>. [cit. 2024-05-05].
- [5] *Akce na klíč: PRAGOKONCERT BOHEMIA*, a.s.. Online. Dostupné z: <https://www.pragokoncert.com/cs/akce-na-klic>. [cit. 2024-05-05].
- [6] *What Is an IDE? : Codecademy*. Online. 2024. Dostupné z: <https://www.codecademy.com/article/what-is-an-ide>. [cit. 2024-05-05].
- [7] *Stack Overflow Developer Survey 2023*. Online. Dostupné z: <https://survey.stackoverflow.co/2023/#most-popular-technologies-new-collab-tools>. [cit. 2024-05-05].
- [8] *Visual Studio 2022 IDE – programovací nástroj pro vývojáře softwaru*. Online. 2024. Dostupné z: <https://visualstudio.microsoft.com/cs/vs/>. [cit. 2024-05-05].
- [9] *What is Database? – GeeksforGeeks*. Online. 2023. Dostupné z: <https://www.geeksforgeeks.org/what-is-database/>. [cit. 2024-05-05]
- [10] SHIELDS, Walter. *SQL QuickStart Guide: The Simplified Beginner's Guide to Managing, Analyzing, and Manipulating Data With SQL*. Albany: ClydeBank Media, c2016. ISBN 9781945051753.
- [11] YASAR, Kinza. *What is Structured Query Language (SQL)?*. Online. In: TechTarget. 2005 – 2024. Dostupné z: <https://www.techtarget.com/searchdatamanagement/definition/SQL>. [cit. 2024-05-05].
- [12] *What Is A Relationship In Database? (Definition And Types) : Indeed.com India*. Online. 2024. Dostupné z: <https://in.indeed.com/career-advice/career-development/what-is-relationship-in-database>. [cit. 2024-05-05]
- [13] *Introduction to NoSQL – GeeksforGeeks*. Online. 2024. Dostupné z: <https://www.geeksforgeeks.org/introduction-to-nosql/>. [cit. 2024-05-05].

- [14] MŮČKA, Jan. *Relační databáze vs. nerelační databáze: Čím se liší?*. Online. In: MasterDC. 27. 5. 2021. Dostupné z: <https://www.master.cz/blog/relacni-databaze-nerelacni-databaze-jake-jsou-rozdily/>. [cit. 2024-05-05].
- [15] *The Different Types of Databases - Overview with Examples*. Online. Dostupné z: <https://www.prisma.io/dataguide/intro/comparing-database-types>. [cit. 2024-05-05].
- [16] *SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS) : Microsoft Learn*. Online. 2024. Dostupné z: <https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>. [cit. 2024-05-05].
- [17] DEAN, John. *Web programming with HTML5, CSS, and JavaScript*. Burlington: Jones & Bartlett Learning, c2019. ISBN 9781284091793.
- [18] *Overview of ASP.NET Core : Microsoft Learn*. Online. 2024. Dostupné z: <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-8.0>. [cit. 2024-05-05].
- [19] *A tour of C# - Overview - C# : Microsoft Learn*. Online. 2024. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/csharp/tour-of-csharp/>. [cit. 2024-05-05].
- [20] PRICE, Mark J.. *C# 11 and .NET 7 - Modern Cross-Platform Development Fundamentals: Start Building Websites and Services with ASP.NET Core 7, Blazor, and EF Core 7*. 7th edition. Birmingham: Packt Publishing, 2022. ISBN 978-1-80323-780-0.
- [21] FINDLAY, Christian. *Blazor Vs. Traditional Web Apps. Blazor*. Online. In: Medium. 9.6.2020. Dostupné z: <https://cfdevelop.medium.com/blazor-vs-traditional-web-apps-8def0aa26c21>. [cit. 2024-05-05].
- [22] *Entity Framework LINQ Queries - Learn the Basics About Querying*. Online. 2023. Dostupné z: <https://entityframework.net/linq-queries>. [cit. 2024-05-05].
- [23] *How do you choose between code first, database first, or model first approaches with EF and MVVM?*. Online. 2024. Dostupné z: <https://www.linkedin.com/advice/1/how-do-you-choose-between-code-first>. [cit. 2024-05-05].
- [24] *Introduction to Identity on ASP.NET Core : Microsoft Learn*. Online. 2024. Dostupné z: <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-7.0&tabs=visual-studio>. [cit. 2024-05-05].

- [25] ANDERSON, David. *Using your own database schema and classes with ASP.NET Core Identity and Entity Framework Core*. Online. In: Danderson. 21.1.2018. Dostupné z: <http://danderson.io/posts/using-your-own-database-schema-and-classes-with-asp-net-core-identity-and-entity-framework-core/>. [cit. 2024-05-05].
- [26] SIMMON, Anthony. *Evolutionary and robust password hashing using PBKDF2 in .NET*. Online. In: Anthonysimmon. 26.12.2023. Dostupné z: <https://anthonysimmon.com/evolutionary-and-robust-password-hashing-using-pbkdf2-in-dotnet/>. [cit. 2024-05-05].
- [27] SVIRCA, Zanfina. *Everything you need to know about MVC architecture*. Online. In: Towards Data Science. 29.5.2020. Dostupné z: <https://towardsdatascience.com/everything-you-need-to-know-about-mvc-architecture-3c827930b4c1>. [cit. 2024-05-05].
- [28] *MVC Design Pattern – GeeksforGeeks*. Online. 2024. Dostupné z: <https://www.geeksforgeeks.org/mvc-design-pattern/#advantages-of-the-mvc-design-pattern>. [cit. 2024-05-05].
- [29] *Overview of ASP.NET Core MVC : Microsoft Learn*. Online. 2024. Dostupné z: <https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0>. [cit. 2024-05-05].
- [30] *About Bootstrap v5.3*. Online. Dostupné z: <https://getbootstrap.com/docs/5.3/about/overview/>. [cit. 2024-05-05].
- [31] HUSSAIN, Kameron a HUSSAIN, Frahaan. *Mastering Bootstrap 5: From Basics to Expert Projects*. Online. Sonar Publishing, 2023. eISBN 9798223722250. Dostupné z: <https://www.everand.com/read/691291772/Mastering-Bootstrap-5-From-Basics-to-Expert-Projects>. [paywall].
- [32] *Bootstrap Grid System – Tutlane*. Online. 2023. Dostupné z: <https://www.tutlane.com/tutorial/bootstrap/bootstrap-grid-system>. [cit. 2024-05-05]
- [33] SHA, Sofia. *Pros Cons Using Bootstrap Web Development*. Online. In: Bootstrap Dash. 24.4.2023. Dostupné z: <https://www.bootstrapdash.com/blog/pros-cons-using-bootstrap-web-development>. [cit. 2024-05-05].
- [34] COLLIER, Michael Collier a SHAHAN, Robin. *Microsoft Azure Essentials: Fundamentals of Azure*. 2nd edition. Microsoft Press, 2016. ISBN 978-1-5093-0296-3.

- [35] *What's the Difference Between AWS vs. Azure vs. Google Cloud? : Coursera*. Online. 2020. Dostupné z: <https://www.coursera.org/articles/aws-vs-azure-vs-google-cloud>. [cit. 2024-05-05].
- [36] *Přehled cen – Od čeho se odvíjejí ceny Azure : Microsoft Azure*. Online. 2024. Dostupné z: <https://azure.microsoft.com/cs-cz/pricing/#product-pricing>. [cit. 2024-05-05].
- [37] *What is azure app service*. Online. 2020. Dostupné z: <https://www.pragimtech.com/blog/azure/what-is-azure-app-service/>. [cit. 2024-05-05].
- [38] IVANOV, Alex. *WYSIWYG Website Builders for Online Business*. Online. In: HuffPost Impact. 15.12.2015. Dostupné z: https://www.huffpost.com/entry/wysiwyg-website-builders_b_8814882. [cit. 2024-05-05].
- [39] *Quill - Your powerful rich text editor*. Online. Dostupné z: <https://quilljs.com/>. [cit. 2024-05-05].
- [40] *FullCalendar - JavaScript Event Calendar*. Online. 2024. Dostupné z: <https://fullcalendar.io/>. [cit. 2024-05-05].
- [41] *Chart.js : Chart.js*. Online. 2024. Dostupné z: <https://www.chartjs.org/docs/latest/>. [cit. 2024-05-05].
- [42] *iText Core: an open-source PDF development library for Java and .NET*. Online. Dostupné z: <https://itextpdf.com/products/itext-core>. [cit. 2024-05-05].
- [43] *Convert HTML & CSS to rich, smart PDF documents – iText*. Online. Dostupné z: <https://itextpdf.com/products/convert-html-css-to-pdf-pdfhtml>. [cit. 2024-05-05].
- [44] *NuGet Gallery : itext7 8.0.4*. Online. 2024. Dostupné z: <https://www.nuget.org/packages/itext7>. [cit. 2024-05-05].
- [45] *Black Box Testing vs. White Box Testing vs. Grey Box Testing – javatpoint*. Online. 2011-2021. Dostupné z: <https://www.javatpoint.com/black-box-testing-vs-white-box-testing-vs-grey-box-testing>. [cit. 2024-05-05].
- [46] *The Four Levels of Software Testing : Segue Technologies*. Online. 2023. Dostupné z: <https://www.seguetech.com/the-four-levels-of-software-testing/>. [cit. 2024-05-05].
- [47] *Key Vault : Microsoft Azure*. Online. 2024. Dostupné z: <https://azure.microsoft.com/cs-cz/products/key-vault>. [cit. 2024-05-05].

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AJAX	Asynchronous JavaScript And XML
AWS	Amazon Web Services
CSS	Cascading Style Sheets
DOM	Document Object Model
EF	Entity Framework
GCP	Google Cloud Platform
GUI	Graphical User Interface
HTML	HyperText Markup Language
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
LINQ	Language Integrated Query
MVC	Model View Controller
ORM	Object-Relational Mapping
PaaS	Platform as a Service
PBKDF2	Password-Based Key Derivation Function 2
PR	Public Relations
RDBMS	Relation Database Management System
SaaS	Software as a Service
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSMS	SQL Server Management Studio
UI	User Interface
URL	Uniform Resource Locator
WYSIWYG	What You See Is What You Get

SEZNAM OBRÁZKŮ

Obrázek 1 Formulář pro vytvoření akce na platformě Facebook [2]	14
Obrázek 2 Nejpoužívanější IDE v roce 2023 podle průzkumu Stack Overflow [7] ..	18
Obrázek 3 Ukázka HTML jazyka	22
Obrázek 4 Ukázka CSS.....	22
Obrázek 5 Ukázka JavaScript jazyka v HTML	23
Obrázek 6 Ukázka syntaxe Blazor [21]	24
Obrázek 7 Ukázka LINQ dotazu	25
Obrázek 8 Výchozí schéma tabulek vygenerovaných Identity [25]	26
Obrázek 9 MVC architektura [29]	28
Obrázek 10 Grid systém frameworku Bootstrap [32].....	29
Obrázek 11 Nejpoužívanější cloudové platformy v roce 2023 podle průzkumu Stack Overflow [7]	30
Obrázek 12 Návrh databáze s relacemi bez Identity	39
Obrázek 13 Ukázka entity CultureEvent	41
Obrázek 14 Ukázka implementace ApplicationDbContext.....	42
Obrázek 15 Implementace navigační lišty v _Layout.cshtml.....	44
Obrázek 16 Navigační lišta nepřihlášeného uživatele	44
Obrázek 17 Navigační lišta přihlášeného uživatele	44
Obrázek 18 Domovská stránka	45
Obrázek 19 Implementace metody Index	46
Obrázek 20 Dynamické zobrazení akcí na hlavní stránce	46
Obrázek 21 Formulář pro přihlášení do Cultur4U	47
Obrázek 22 Verifikační obrazovka	48
Obrázek 23 Ukázka implementace služby EmailSender	49
Obrázek 24 Ukázka příchozí verifikační e-mailové zprávy	49
Obrázek 25 Modální okno pro vytvoření závěrečné zprávy.....	50
Obrázek 26 Návrh závěrečné zprávy	51
Obrázek 27 Ukázka části kódu implementace generování závěrečné zprávy	52
Obrázek 28 Ukázka přepínání mezi jednotlivými záložkami v detailu společnosti ...	53
Obrázek 29 Javascript funkce pro přidání člena do společnosti	54
Obrázek 30 Implementace metody AddUserToCompany	54
Obrázek 31 Ukázka zobrazené notifikační zprávy	55

Obrázek 32 Implementace notifikační zprávy	55
Obrázek 33 Nastavení uživatele	56
Obrázek 34 Ukázka obrazovky při nedostatečných právech uživatele.....	56
Obrázek 35 Implementace kontroly práv uživatele	57
Obrázek 36 Implementace WYSIWYG editoru s toolbarem	58
Obrázek 37 Formulář pro vytvoření nové akce	58
Obrázek 38 Implementace metody SaveFileAsync	59
Obrázek 39 Implementace služby BlobService	60
Obrázek 40 Implementace metody pro mazání souborů	61
Obrázek 41 Graf zájmu uživatelů	61
Obrázek 42 Implementace kódu pro statistiky akce	62
Obrázek 43 Vizualizace účasti na akci pomocí Chart.js.....	63
Obrázek 44 Zobrazení kalendáře s akcemi	64
Obrázek 45 Mobilní zobrazení domovské stránky	66
Obrázek 46 Mobilní zobrazení stránky EventList	66
Obrázek 47 Ukázka implementace jednotkového testu	67
Obrázek 48 Modální okno pro vytvoření nové App Service	68
Obrázek 49 Možnost povolení migrace při publikaci.....	69

SEZNAM PŘÍLOH

PŘÍLOHA P I: Detail akce z pohledu organizátora

PŘÍLOHA P II: Vygenerovaná závěrečná zpráva

PŘÍLOHA P III: CD

PŘÍLOHA P I: DETAIL AKCE Z POHLEDU ORGANIZÁTORA

Celestial Symphony: An Illuminated Music Festival

20 May 2024 - 23 May 2024

Tickets

- Day pass - 50 \$
- Full Festival Pass (3 days) - 120 \$
- VIP Pass (Meet & Greet, free refreshments) - 200 \$
- Group Discount (10+ people) - 10 \$

Location

Lakeside Amphitheater

Contact

galaxyoflights@galaxy.com

+44 1122 334455



Important Description:

Tickets must be purchased online and in advance

The festival runs rain or shine; please come prepared for all weather conditions

An array of food and drink options will be available from local and gourmet vendors

On-site free parking is available

The event is family-friendly, but children under 16 must be accompanied by an adult

Please respect the natural setting and participate in our recycling and sustainability efforts

I will attend

I am interested

Customize PDF Report



Description

Celestial Symphony is a groundbreaking music festival presented by Galaxy of Lights, where the harmony of sound meets the radiance of light in an unforgettable outdoor setting.

This event combines **live music performances** from a diverse lineup of artists and bands **with an extraordinary light show**, creating an immersive audio-visual experience. Attendees will journey through **different music genres** each enhanced with its own unique light display.

- classical to contemporary
- jazz to electronic

Imagine orchestras playing under a canopy of stars, DJs spinning tracks amidst lasers and neon sculptures, and bands performing with the backdrop of interactive light installations that respond to the rhythm of the music.

Program Highlights:

- May 20, 2024 - Opening Night Spectacular**
- 4:00 PM: Gates Open - Start your journey with ambient sounds and gentle lights.
- 5:30 PM: Emerging Artists Showcase - Experience fresh talents under a twilight glow.
- 8:00 PM: Light Parade - A procession of illuminated performers sets the night ablaze.
- 9:00 PM: Headliner Act - A world-renowned band performs, synchronized with an exclusive light show.
- May 21, 2024 - Genre Journey**
- 4:00 PM: Gates Open - Ambient sessions resume with interactive light sculptures.
- 5:00 PM: Jazz Under the Stars - Smooth jazz sets accompanied by soft, immersive lighting.
- 7:30 PM: Electronic Odyssey - DJs and electronic acts light up the night with beats and lasers.
- 9:30 PM: Rock Rhapsody - Rock bands electrify the stage with dynamic light effects.
- May 22, 2024 - Finale Fusion**
- 4:00 PM: Gates Open - The final day begins with an ambient and tranquil light setting.
- 6:00 PM: Classical Crossover - A blend of classical music with modern light designs.
- 8:30 PM: Global Beats - World music artists perform amidst a kaleidoscope of lights.
- 11:00 PM: Grand Finale - A spectacular closing act followed by an unforgettable light and firework show.

Photo Gallery



Video Gallery



PŘÍLOHA P II: VYGENEROVANÁ ZÁVĚREČNÁ ZPRÁVA

Celestial Symphony: An Illuminated Music Festival

20 May 2024 - 23 May 2024

Lakeside Amphitheater

Galaxy of Lights

Galaxy of Lights is an innovative event company that specializes in creating immersive light-based experiences and cultural events that enchant and inspire. With a decade of experience in blending art, technology, and light, the company has become a pioneer in producing events that transform public spaces into dazzling realms of color, light, and imagination. Galaxy of Lights is committed to sustainability and uses energy-efficient lighting technology to minimize environmental impact.

Mobile: +44 1122 334455

Email: galaxyoflights@galaxy.com

Facebook: <https://facebook.com/galaxyoflights>

Instagram: <https://instagram.com/galaxyoflightsoffical>

Website: <https://galaxyoflights.website.com>



Description:

Celestial Symphony is a groundbreaking music festival presented by Galaxy of Lights, where the harmony of sound meets the radiance of light in an unforgettable outdoor setting.

This event combines **live music performances** from a diverse lineup of artists and bands **with an extraordinary light show**, creating an immersive audio-visual experience. Attendees will journey through **different music genres** each enhanced with its own unique light display.

1. classical to contemporary
2. jazz to electronic

Imagine orchestras playing under a canopy of stars, DJs spinning tracks amidst lasers and neon sculptures, and bands performing with the backdrop of interactive light installations that respond to the rhythm of the music.

Program Highlights:

1. May 20, 2024 - Opening Night Spectacular

2. 4:00 PM: Gates Open - Start your journey with ambient sounds and gentle lights.
3. 5:30 PM: Emerging Artists Showcase - Experience fresh talents under a twilight glow.
4. 8:00 PM: Light Parade - A procession of illuminated performers sets the night ablaze.
5. 9:00 PM: Headliner Act - A world-renowned band performs, synchronized with an exclusive light show.

6. May 21, 2024 - Genre Journey

7. 4:00 PM: Gates Open - Ambient sessions resume with interactive light sculptures.
8. 5:00 PM: Jazz Under the Stars - Smooth jazz sets accompanied by soft, immersive lighting.
9. 7:30 PM: Electronic Odyssey - DJs and electronic acts light up the night with beats and lasers.
10. 9:30 PM: Rock Rhapsody - Rock bands electrify the stage with dynamic light effects.

11. May 22, 2024 - Finale Fusion

12. 4:00 PM: Gates Open - The final day begins with an ambient and tranquil light setting.
13. 6:00 PM: Classical Crossover - A blend of classical music with modern light designs.
14. 8:30 PM: Global Beats - World music artists perform amidst a kaleidoscope of lights.
15. 11:00 PM: Grand Finale - A spectacular closing act followed by an unforgettable light and firework show.



Final Report:

The Celestial Symphony: An Illuminated Music Festival, brilliantly crafted by Galaxy of Lights, took the city by storm from May 20th to 23rd, 2024, at the picturesque Lakeside Amphitheater. This event merged the rhythmic beauty of diverse musical genres with mesmerizing light shows, crafting an unforgettable spectacle that drew an estimated audience of over 10,000 people each day, ranging from avid music lovers to casual festival-goers seeking a unique experience. As the gates opened each evening, attendees were welcomed into a transformed world where the harmony of sound met the radiance of light. The festival kicked off with the "Opening Night Spectacular," setting a high bar for the nights to follow. Each evening was a journey through different musical landscapes, from the soothing melodies of jazz under starlit skies to the vibrant beats of electronic dance music pulsing through the crowd. The light parade was a highlight, featuring illuminated performers that lit up the night, blending seamlessly with live music to enhance the sensory experience of the attendees.

The "Genre Journey" night saw genres blending at each turn of the corner, with orchestras playing classical tunes that morphed into contemporary beats, all accompanied by bespoke light designs that shifted with the music's mood. The "Finale Fusion" was a triumph of both auditory and visual spectacle, culminating in a grand finale that combined live performances with a stunning fireworks display, leaving the audience in awe.

Success of the Celestial Symphony was not just in its execution but also in its ability to attract a diverse audience, thanks to Galaxy of Lights' commitment to creating inclusive and accessible cultural experiences. Families, young adults, and older attendees alike found something to cherish, from the Emerging Artists Showcase, which spotlighted new talents, to the sophisticated setups allowing seamless movement for all, including those with mobility challenges. The social media buzz created by the event was monumental, with thousands of posts, shares, and likes across platforms like Instagram and Facebook, further amplified by the engaging content shared by Galaxy of Lights.

Overall, the Celestial Symphony: An Illuminated Music Festival was a resounding success, a testament to the power of combining art and technology to create magical experiences. The event set a new standard for music festivals and highlighted Galaxy of Lights' prowess in event organization and innovation. This festival has undoubtedly left an indelible mark on its participants, promising even greater spectacles in the future.

PŘÍLOHA P III: CD

CD obsahuje:

- Zdrojové kódy aplikace Cultur4U: prilohy.zip
- Bakalářská práce ve formátu PDF: fulltext.pdf