



# SPOVYZ

Dokumentace k ročníkové práci

**Autor:** Jan Vacek

**Třída:** 3ITB

**Vedoucí práce:** Tomáš Ulrich

2024/2025



## Prohlášení

---

Prohlašuji, že jsem ročníkovou práci na téma „Spovyz“ vypracoval samostatně a s použitím uvedené literatury a pramenů.

V (Trmicích) dne .....

.....

## Poděkování

---

Chtěl bych poděkovat Tomáši Ulrichovi za vedení mé ročníkové práce a odborný dohled nad prací. Dále bych chtěl poděkovat Bc. Vratislavovi Medřickému za cenné odborné rady při vývoji webové aplikace, Rudovi Dařílkovi za odborné rady s pozicováním prvků za použití css, mamince Mgr. Evě Vackové za pomoc při gramatické kontrole dokumentace a Bc. Kateřině Kubů za psychickou podporu při vývoji celého projektu.

# Anotace

---

Tento dokument je rozvržen na šest hlavních částí. Úvod, kde popisuji záměr ročníkové práce a to, proč jsem si vybral dané téma. Druhá je teoretická část. Ta se dále dělí na řešerši, kde uvádím informace o mé inspiraci podpořené výzkumy a studiemi, a technologie, které jsem při vývoji webové aplikace použil. Další velká část je praktická. Dělí se na návrhy, produktizaci a popis pro uživatele. V návrzích popisuji moje začátky s prací a mé původní plány pro vývoj. Produktizace ukazuje důležité programové části aplikace a popis pro uživatele zobrazuje návod na použití aplikace. Použitá literatura obsahuje odkazy na všechny využití technologie a informace použité při práci na projektu. Poslední částí je seznam obrázků, kde se nachází popisky všech obrázků použitých v dokumentaci.

## Klíčová slova

---

Webová aplikace, Spolupráce, Vykazování, Firma, Projekt, Task, Stres, Klidné prostředí, Intuitivní, Minimalismus

## Obsah

Úvod .....	8
1 Teoretická část .....	9
1.1 Rešerše.....	9
1.1.1 Inspirace .....	9
1.1.2 Výzkumy a studie .....	10
1.2 Technologie .....	12
1.2.1 C# .....	12
1.2.2 Bootstrap .....	12
1.2.3 Npgsql .....	12
1.2.4 Microsoft Entity Framework .....	12
1.2.5 Bootstrap Studio .....	12
1.2.6 Visual Studio.....	12
1.2.7 Visual Studio Code .....	13
1.2.8 AI Nero .....	13
1.2.9 Gemini.....	13
1.2.10 Figma.....	13
1.2.11 Nebo.....	13
1.2.12 Draw.io .....	14
1.2.13 Photopea.....	14
2 Praktická část.....	15
2.1 Návrhy.....	15
2.1.1 Výběr technologie serveru .....	15
2.1.2 Návrh propojení všech stránek webu v Nebo.....	15
2.1.3 Návrh designu webu ve Figmě.....	16
2.1.4 Návrh databáze v Nebo.....	16
2.1.5 Návrh diagramu databáze pomocí Draw.io .....	17
2.2 Produktizace .....	18
2.2.1 Vývoj v Bootstrap Studiu.....	18
2.2.2 Generování hash pro heslo .....	18
2.2.3 Cors Policy .....	19
2.2.4 Připojení na server pomocí JavaScript .....	19
2.2.5 Tabulka pomocí Microsoft Entity Frameworku .....	20
2.2.6 Použití Microsoft Entity Frameworku .....	20

2.2.7	Původní implementace kontroleru .....	21
2.2.8	Použitá implementace kontroleru .....	21
2.2.9	Implementace servisu .....	22
2.2.10	Implementace repositář.....	23
2.2.11	Práva přístupu ke kontroleru .....	23
2.2.12	Validace dat.....	23
2.2.13	Logo.....	24
2.3	Popis pro uživatele.....	24
Použitá literatura.....		26
Seznam obrázků .....		28
Seznam kódů .....		29
Obsah média .....		30

# Úvod

---

Cílem této ročníkové práce je vytvořit webovou aplikaci, která výrazně zjednoduší spolupráci na projektech v rámci firmy. Aplikace bude sloužit jako centralizované místo pro správu úkolů, projektů a vykazování odvedené práce. Hlavní důraz bude kladen na vytvoření intuitivního a přehledného uživatelského rozhraní, které minimalizuje stres a usnadňuje orientaci v aplikaci. Díky minimalistickému designu se uživatelé budou moci soustředit na samotnou práci a nebudou rozptylováni zbytečnými vizuálními prvky.

Aplikace bude implementovat komplexní systém oprávnění, který zajistí přehlednou a bezpečnou správu přístupu k různým funkcionalitám. Uživatelé budou rozděleni do několika úrovní s jasně definovanými rolemi. Majitel bude mít nejvyšší úroveň přístupu, avšak pouze v režimu zobrazení, což mu umožní získat komplexní přehled o všech aktivitách v aplikaci, ale nebude moci provádět žádné změny. Administrátor bude zodpovědný za správu uživatelských účtů, přiřazování rolí a správu oprávnění. Ředitelé budou mít přehled o všech projektech, jejich stavu a finančních aspektech. Vedoucí budou spravovat jednotlivé projekty a přidělovat úkoly svým týmům. Zaměstnanci budou přímo vykonávat zadané úkoly a zaznamenávat svůj pokrok. Ekonom bude zodpovědný za plánování a řízení finančních aktivit společnosti a bude mít komplexní přehled o všech finančních záležitostech.

Pro zajištění co nejlepšího přehledu o průběhu projektů, budou všechny projekty rozděleny na jednotlivé úkoly. Každý úkol bude obsahovat veškeré potřebné informace, jako je popis, termín dokončení, zodpovědná osoba a aktuální stav. Tento detailní popis umožní snadno sledovat pokrok každého úkolu a přehled o celém projektu

Velký důraz bude kladen na vytvoření příjemného a intuitivního uživatelského prostředí, které bude podporovat psychickou pohodu zaměstnanců. Minimalistický design a přehledné rozhraní pomohou snížit stres a usnadní orientaci v aplikaci. Cílem je vytvořit prostředí, ve kterém se zaměstnanci budou cítit příjemně a budou moci efektivně pracovat.

Největší inspirací pro mě je aplikace Azure DevOps, která se zaměřuje na správu projektů, mzdový systém Vema a webová stránka Calm, která se formou blogů zaměřuje na psychické zdraví.

Vývoj této webové aplikace jsem si zvolil z důvodu velkého využití a budoucnosti webových aplikací. Chci si vyzkoušet a zdokonalit se při tvorbě API. Hlavní motivací je dát dohromady můj zájem o pracovní aplikace pro firmy a zájem o psychologii se zaměřením na psychickou pohodu zaměstnanců.



# 1 Teoretická část

---

## 1.1 Rešerše

---

### 1.1.1 Inspirace

#### Azure DevOps

Azure DevOps je sada webových aplikací, která poskytuje komplexní sadu nástrojů pro plánování, vývoj, testování, nasazování a monitorování softwarových projektů. Nabízí integraci správy verzí (Azure Repos), sledování práce (Azure Boards), CI/CD pipelines (Azure Pipelines) a testování (Azure Test Plans). S Azure DevOps můžou týmy efektivně spolupracovat, automatizovat procesy a rychleji dodávat kvalitní software. (1)

Azure DevOps mě zaujal svou komplexností v oblasti řízení projektů. Avšak jeho primární zaměření na vývojáře limituje jeho využití pro širší spektrum profesí. Cílem mého projektu je vytvořit aplikaci, která by byla užitečná pro různé obory.

#### Vema

Mzdový systém Vema je komplexní a spolehlivé řešení pro zpracování mezd a personalistiku v České republice. Nabízí širokou škálu funkcí, od výpočtu mezd a daní, přes správu docházky a absencí, až po generování reportů a komunikaci s úřady. Systém je neustále aktualizován dle platné legislativy a je navržen tak, aby zjednodušil a zefektivnil mzdovou agendu firem všech velikostí a zaměření. (2)

Na systému Vema mě inspirovala správa docházky a výpočty mezd. Tuto mechaniku bych proto chtěl začlenit do svého projektu.

#### Calm

Calm je webová stránka a mobilní aplikace zaměřená na podporu duševního zdraví a well-beingu. Nabízí širokou škálu meditačních cvičení, relaxačních technik, programů pro zlepšení spánku a dalšího obsahu, který pomáhá uživatelům snižovat stres, úzkost a zlepšovat celkovou psychickou pohodu. Calm je dostupný v mnoha jazycích a nabízí personalizovaná doporučení pro individuální potřeby uživatelů. (3)

Stránka Calm mě zaujala svým zaměřením na duševní zdraví. Inspiroval jsem se zde nejen výběrem barev, ale také tvary vizuálních prvků stránky, protože jedním z hlavních cílů stránky je, aby uživatelé přivedla do psychického klidu a duševní pohody.

### 1.1.2 Výzkumy a studie

Tyto zdroje informací mi poskytly data, na jejichž základě jsem zvolil výběr barev a vzhled uživatelského rozhraní mé aplikace. Většina dat analyzuje pracoviště ve fyzické podobě nebo mezilidské vztahy. Jelikož softwarové prostředí je stejně důležité jako to fyzické, řídil jsem se při vývoji těmito získanými informacemi.

#### Psychologie barev v marketingu: Význam barev?

Stránka se zabývá psychologií barev v marketingu a jejich významem pro značky a produkty. Popisuje, jak různé barvy ovlivňují podvědomí lidí a jak mohou vyvolávat specifické emoce a reakce. Například modrá barva symbolizuje jistotu a bezpečí, zatímco červená vzbuzuje vášně a energii. Stránka také uvádí příklady, jak firmy využívají barvy ve svých logách a reklamách k dosažení lepšího marketingového efektu. (4)

#### Vhodné pracovní prostředí jako klíč k produktivitě a zdraví zaměstnanců

Stránka se zabývá významem zdravého pracovního prostředí pro produktivitu a zdraví zaměstnanců. Zdůrazňuje, že vhodné pracovní podmínky, jako je kvalitní osvětlení, dobrá akustika a dostatek čerstvého vzduchu, mohou výrazně přispět k prevenci nemocí a zlepšení celkové pohody zaměstnanců. Dále se zaměřuje na důležitost prevence a zdravého životního stylu, který může snížit výskyt závažných onemocnění. Stránka také upozorňuje na to, že podpora zdraví by měla být součástí firemní kultury a že kvalitní pracovní prostředí může zvýšit motivaci a produktivitu zaměstnanců. (5)

#### Ovlivňuje vzhled pracoviště produktivitu?

Stránka se zabývá estetickým vzhledem pracoviště, který má významný vliv na produktivitu a spokojenost zaměstnanců. Dobře navržený pracovní prostor, který zahrnuje přirozené světlo, barvy, rostliny a ergonomický nábytek, může zvýšit motivaci, snížit stres a podpořit kreativitu. Esteticky příjemné prostředí také přispívá k lepšímu náboru a udržení zaměstnanců, protože lidé preferují pracovat v harmonickém a inspirativním prostředí. Společnosti jako Google a Facebook jsou lídry v oblasti designu kanceláří, což ukazuje, že investice do estetického vzhledu pracoviště se vyplatí. (6)

#### 4 z 10 mladých Čechů v posledním roce absentovali v práci kvůli stresu

Stránka se zabývá výsledky mezinárodního výzkumu o vnímání duševního zdraví, který realizovala společnost Ipsos. Výzkum ukazuje, že 4 z 10 mladých Čechů v posledním roce absentovali v práci kvůli stresu. Celkově se 6 z 10 Čechů potýkalo se stresem, který významně ovlivnil jejich běžný život. Mladí lidé do 30 let pociťují stres nejvíce a často kvůli němu chybí v práci. Výzkum také upozorňuje na nerovný přístup zdravotnických systémů k duševnímu zdraví a na to, že veřejnost vnímá, že fyzické zdraví je upřednostňováno před duševním. (7)

Jak je na tom psychika mladých dospělých? Vědci ji v unikátním výzkumu studovali deset let.

Stránka se zabývá desetiletým výzkumem duševního zdraví mladých dospělých ve věku 18 až 30 let, který prováděli vědci z Psychiatrické kliniky 1. LF UK a VFN. Výzkum ukazuje, že moderní životní styl, digitální média a sociální sítě zvyšují stres, úzkost a pocit izolace. Zjistilo se, že úzkost, deprese a syndrom vyhoření jsou častější u žen než u mužů, a že špatný spánek a denní konzumace alkoholu zhoršují příznaky. Výzkum zdůrazňuje potřebu sledovat duševní zdraví mladých dospělých a integrovat psychologickou podporu do vzdělávacích a profesních programů, aby se zlepšila kvalita jejich života a podpořila odolnost společnosti. (8)

Pracovní podmínky a jejich vliv na výkon zaměstnance – Hana Kliková

Tato bakalářská práce se zabývá pracovními podmínkami a jejich dopadem na spokojenost a výkon zaměstnanců. Práce zkoumá jak fyzické, tak sociálně-psychologické faktory, které ovlivňují pracovní prostředí a výkonnost. V praktické části práce bylo provedeno dotazníkové šetření mezi zaměstnanci Magistrátu města Plzně, zaměřené na zjišťování jejich spokojenosti s pracovními podmínkami a jejich vnímání stresu. Výsledky šetření ukázaly, že většina zaměstnanců je spokojena s pracovními podmínkami, ale existují i oblasti, které vyžadují zlepšení, například otevřenost komunikace a vnímání stresu. Práce doporučuje, aby vedení magistrátu věnovalo pozornost těmto oblastem a zavedlo opatření ke zlepšení spokojenosti a výkonnosti zaměstnanců. (9)

Vliv pracovního prostředí na výkonnost pracovníků – Šárka Kmoníčková

Tato bakalářská práce zdůrazňuje negativní vliv stresu na výkonnost pracovníků. Popisuje různé stresory, které se mohou vyskytovat na pracovišti, a uvádí jejich projevy a dopady na psychický a fyzický stav zaměstnanců. Mezi nejčastější stresory patří pracovní přetížení, konflikty, nejistota a špatné mezilidské vztahy. Práce dále uvádí, že stres může vést ke snížení výkonnosti, zhoršení koncentrace a paměti, a v extrémních případech až k chronické únavě a depresi. (10)

## 1.2 Technologie

---

### 1.2.1 C#

Jazyk C# je multiplatformní, objektově orientovaný a typově bezpečný jazyk pro obecné účely, který vývojářům umožňuje produktivní práci při psaní vysoce výkonného kódu. C# je nejoblíbenější jazyk pro vývoj pro platformu .NET. (11)

Tento jazyk používám s použitím frameworku ASP .NET Core k psaní kódu API, které funguje jako server pro můj web.

### 1.2.2 Bootstrap

Bootstrap je populární framework pro front-end vývoj. Poskytuje hotové komponenty (tlačítka, navigace, mřížky atd.), které lze jednoduše přizpůsobit jakýmkoliv potřebám. Jeho největší výhodou je předpřipravená responzivita prvků. (12)

Bootstrap používám pro vytváření stylu prvků a responzivního vzhledu webu.

### 1.2.3 Npgsql

Npgsql je open source poskytovatel dat .NET pro PostgreSQL. Umožňuje připojit se a komunikovat se serverem PostgreSQL pomocí .NET. (13)

Tuto knihovnu používám na serverové části pro komunikaci s databází PostgreSQL.

### 1.2.4 Microsoft Entity Framework

Entity Framework je open-source ORM (Object-Relational Mapper) framework pro .NET. To umožňuje mapovat objekty v aplikaci na tabulky v databázi, díky čemuž lze pracovat s daty na vyšší úrovni abstrakce a tím zjednodušuje vývoj aplikací a zvyšuje bezpečnost komunikace s databází. (14)

Entity framework používám k vytvoření struktury databáze, díky čemuž jsem mohl všechny tabulky definovat jako třídy v C#.

### 1.2.5 Bootstrap Studio

Bootstrap Studio je revoluční nástroj pro tvorbu webových stránek pro vizuální navrhování webových stránek a aplikací. Webové stránky, jež vytvoříte v Bootstrap Studiu, se načítají bleskově, jsou optimalizovány pro SEO a jsou plně responzivní. (15)

Bootstrap Studio využívám k vytváření designu celého webu pomocí grafického návrháře.

### 1.2.6 Visual Studio

Visual Studio je integrované vývojové prostředí (IDE), které poskytuje vývojářům širokou škálu nástrojů pro tvorbu softwaru. Nabízí zvýraznění syntaxe, doporučení kódu, grafické editory a obsahuje

také debugger pro snadnou opravu chyb v kódu. Podporuje programování v mnoha různých jazycích, jako je C#, F#, C++, Visual Basic a Python. (16)

Visual Studio používám pro psaní kódu mého API.

#### 1.2.7 Visual Studio Code

Visual Studio Code je bezplatný open-source editor kódu. Podporuje mnoho jazyků, včetně JavaScriptu, Pythonu, Javy a mnoha dalších. Díky obrovskému množství dostupných rozšíření je velmi oblíbený pro vývoj webových aplikací. Pro snadné ovládání verzí kódu je v editoru integrovaný Git. (17)

Visual Studio Code používám pro psaní JavaScript kódu na mém webu.

#### 1.2.8 AI Nero

AI Nero je umělá inteligence, která přetvoří vložené fotografie a obrázky v jasnější, ostřejší a s větším rozlišením. (18)

AI Nero používám pro vylepšování kvality obrázků pro můj web.

#### 1.2.9 Gemini

Gemini je pokročilá umělá inteligence vyvinutá společností Google. Je schopna generovat text, překládat jazyky, psát různé druhy kreativního obsahu a odpovídat na otázky. (19)

Gemini využívám ke generování obrázků pro můj web a také pro řešení problémů s technologiemi, se kterými se učím pracovat.

#### 1.2.10 Figma

Figma je webová aplikace pro grafický design a prototypování, která umožňuje designérům spolupracovat v reálném čase na tvorbě uživatelských rozhraní, webových stránek, mobilních aplikací a dalších grafických prvků. Nabízí intuitivní rozhraní s širokou škálou nástrojů pro vektorovou grafiku, prototypování a sdílení návrhů. Díky cloudovému úložišti a funkcím pro spolupráci je Figma oblíbeným nástrojem pro designéry i vývojáře, kteří chtějí efektivně spolupracovat na projektech a vytvářet moderní a responzivní designy. (20)

Figmu jsem použil na hrubý návrh uživatelského rozhraní mé webové aplikace.

#### 1.2.11 Nebo

Nebo je aplikace pro vytváření ručně psaných poznámek a profesionálních dokumentů, skicování na neomezené plátno a komentování PDF souborů. Disponuje funkcí převodu rukopisu na text, podporu gest a kreslení dokonalých tvarů. To vše pomocí umělé inteligence. (21)

Aplikaci Nebo používám pro téměř všechny návrhy mé webové aplikace.

#### 1.2.12 Draw.io

Draw.io (diagrams.net) je bezplatná online aplikace pro tvorbu diagramů, která nabízí širokou škálu funkcí a možností. Umožňuje snadno vytvářet různé typy diagramů, včetně vývojových diagramů, UML diagramů, ER diagramů, síťových diagramů a mnoho dalších. Draw.io se vyznačuje intuitivním uživatelským rozhraním, rozsáhlou knihovnou tvarů a šablon a možností importu a exportu v různých formátech. Díky své dostupnosti online a integraci s cloudovými úložišti je draw.io ideálním nástrojem pro spolupráci na diagramech v rámci týmu. (22)

Draw.io jsem použil pro grafický návrh databáze pro svůj projekt.

#### 1.2.13 Photopea

Photopea je online grafický editor, který umožňuje úpravu rastrových i vektorových obrázků přímo v prohlížeči. Podporuje formáty jako PSD, XCF, Sketch, AI a další, což z něj činí dobrou alternativu k Adobe Photoshopu. Nabízí pokročilé nástroje pro práci s vrstvami, maskami, textem i filtry. Je zdarma a dostupný bez nutnosti instalace. (23)

Webovou aplikaci Photopea jsem použil pro tvorbu loga.

## 2 Praktická část

Všech návrhů, které jsem udělal, jsem se při vývoji držel. Avšak před zahájením programování jsem udělal drobné či více významné návrhové chyby, takže finální verze se od návrhů mírně či více liší.

### 2.1 Návrhy

#### 2.1.1 Výběr technologie serveru

Při výběru jsem napsal dotaz umělé inteligenci Gemini, které technologie bych mohl zvolit pro vývoj serveru pro moji webovou aplikaci. Rozhodoval jsem se mezi použitím Node.js + Express.js, PHP s Laravel frameworkem a C# ASP.NET API frameworkem.

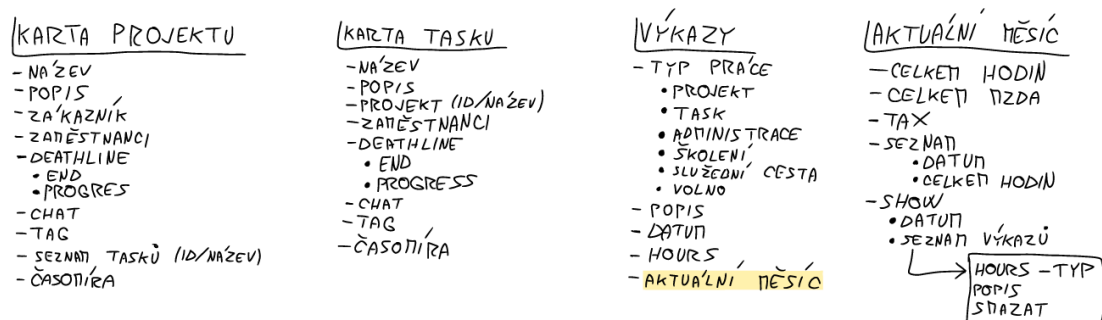
Základy JavaScriptu, PHP i C# umím. Jen jsem nikdy nepracoval s výše vypsány mi frameworky. Všechny jsou v komerčním prostředí hojně využívány. Mou volbou se nakonec stalo použití C# ASP.NET API.

Rozhodl jsem se tak proto, že C# mi ze všech těchto technologií vyhovuje nejvíce. Vytvořit funkční API pro mě bylo cenným získáním nových zkušeností, které mohu v budoucí profesi využít.

Při použití C# jsem ke komunikaci s databází použil Microsoft Entity Framework, objektově orientovaný přístup k databázi pomocí C# tříd, namísto použití SQL dotazů. Práce s Microsoft Entity Frameworkem je pro zaměstnavatele při vybírání budoucího zaměstnání žádanou zkušeností.

#### 2.1.2 Návrh propojení všech stránek webu v Nebo

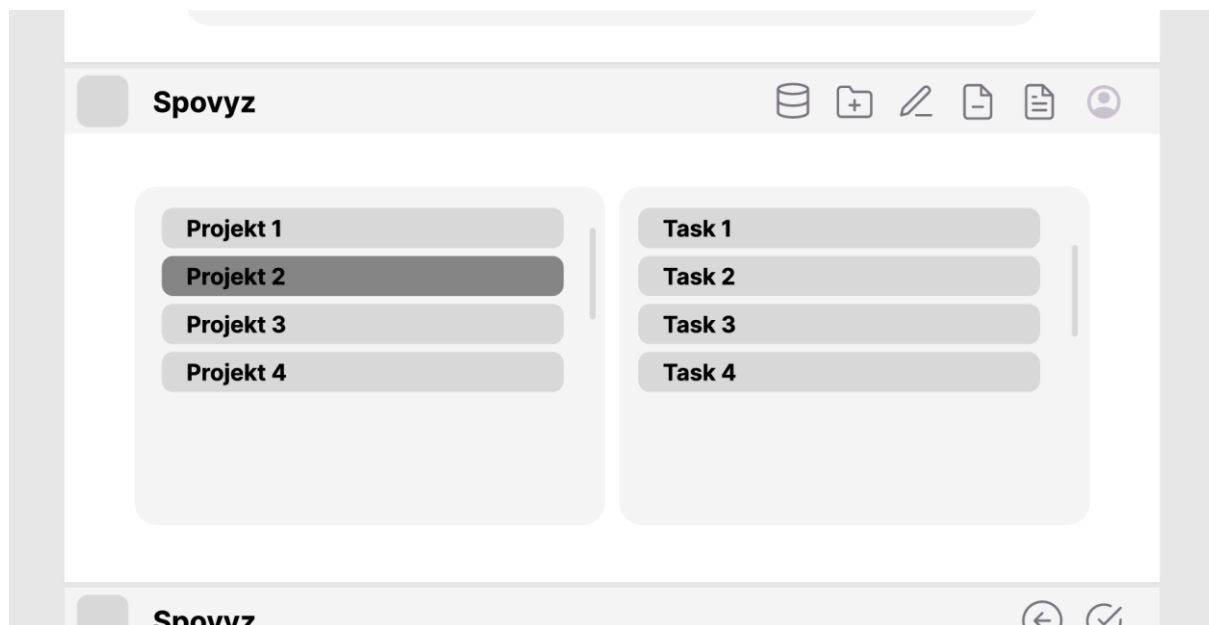
Pro všechny stránky webu jsem navrhnul názvy a vytvořil k nim podrobnou specifikaci. Tato specifikace zahrnuje požadavky na textové vstupy, které uživatelé budou moci vyplňovat, a popis interaktivních prvků jako jsou tlačítka, odkazy nebo formuláře, které budou na stránkách umístěny a umožní uživatelům s webem interagovat.



Obrázek 1: Část návrhu propojení všech stránek webu v Nebo

### 2.1.3 Návrh designu webu ve Figmě

Podle vytvořeného návrhu specifikací jsem navrhl předběžný vzhled uživatelského rozhraní pro všechny stránky webu. Grafický návrh nezahrnuje výběr barev, konkrétních ikon a dalších vizuálních prvků. Udělal jsem pouze surový návrh stránky s prvky, které společně vytvářejí jednotný a estetický dojem, abych měl představu, jak aplikace může vypadat a jakým směrem se při vývoji ubírat. V rámci toho jsem také navrhl layout jednotlivých stránek, včetně umístění jednotlivých prvků a jejich vzájemných vztahů, které již tvořili základ pro vývoj.



Obrázek 2: Část návrhu designu ve Figmě

### 2.1.4 Návrh databáze v Nebo

Stejně jako jsem vytvořil textový návrh pro všechny stránky, vytvořil jsem také návrh pro všechny tabulky v databázi s jejich specifikacemi. Během programování jsem musel původní návrhy databáze v jistých oblastech změnit, protože mi nevyhovovaly nebo jsem na některé důležité specifikace při návrhu zapomněl.

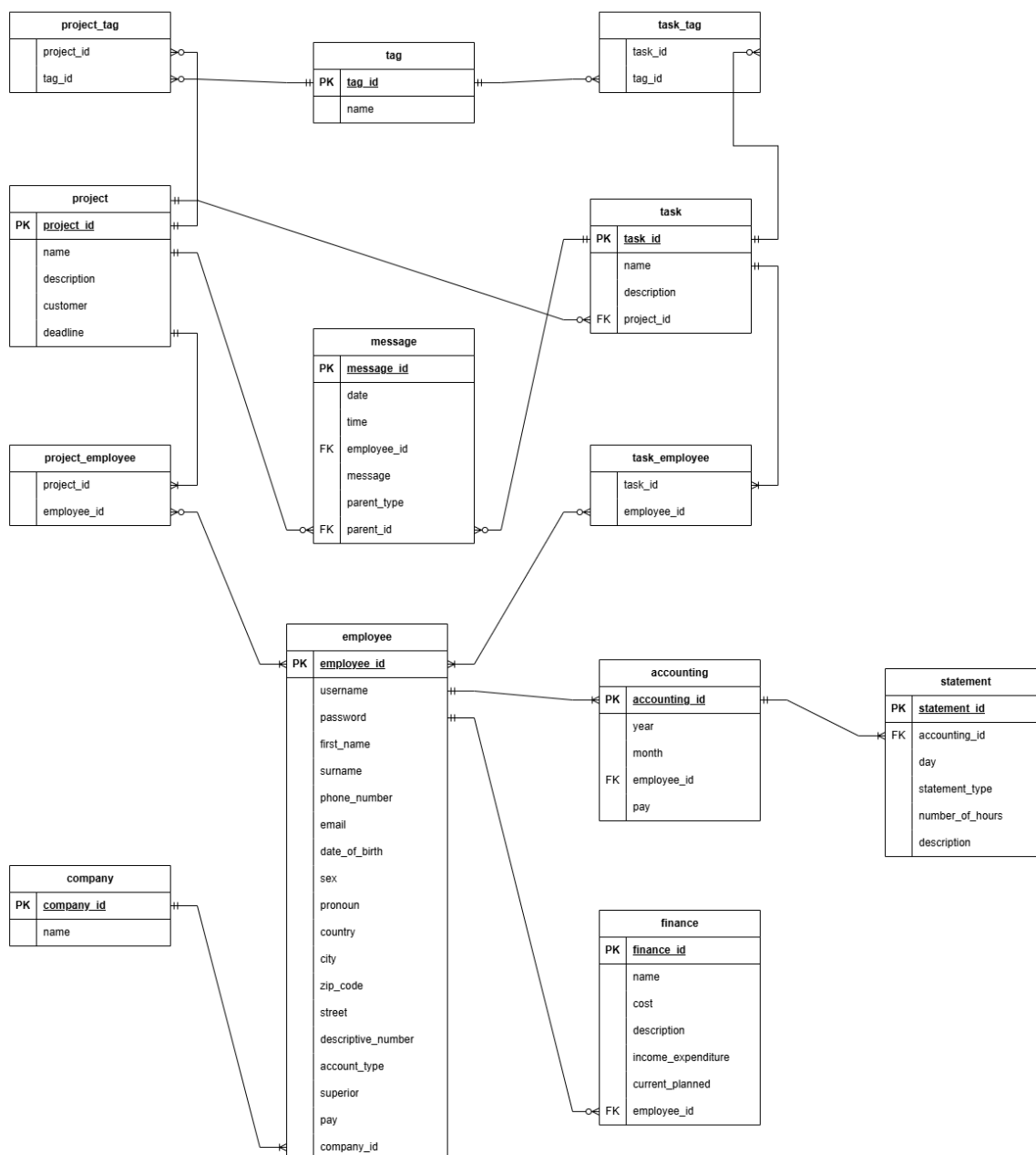
CHAT	VÝKAZ	ÚČTOVÁNÍ	FINANCE
PK - ID	PK - ID	PK - ID	PK - ID
- DATUM	FK - ÚČTOVÁNÍ - ID	- ROK } DATUM	- NÁZEV
- ČAS	- DEN	- MĚSÍC }	- CENA
- ZAMĚSTNANEC	E - TYP	FK - ZAMĚSTNANEC	- POPIS
- MESSAGE	- POČET HODIN	- MZDA	- PŘÍJEM / VÝDAJ
FK - ODKAZ - ID	- POPIS		- AKTUÁLNÍ / PLÁNOVANÝ
E - ODKAZ_TYP			FK - ZAMĚSTNANEC - ID

Obrázek 3: Část návrhu databáze v Nebo



### 2.1.5 Návrh diagramu databáze pomocí Draw.io

Podle textového návrhu databáze jsem následně vytvořil grafický návrh diagramu. Tento diagram sloužil pro základní orientaci v databázi a zobrazení všech vztahů mezi tabulkami.

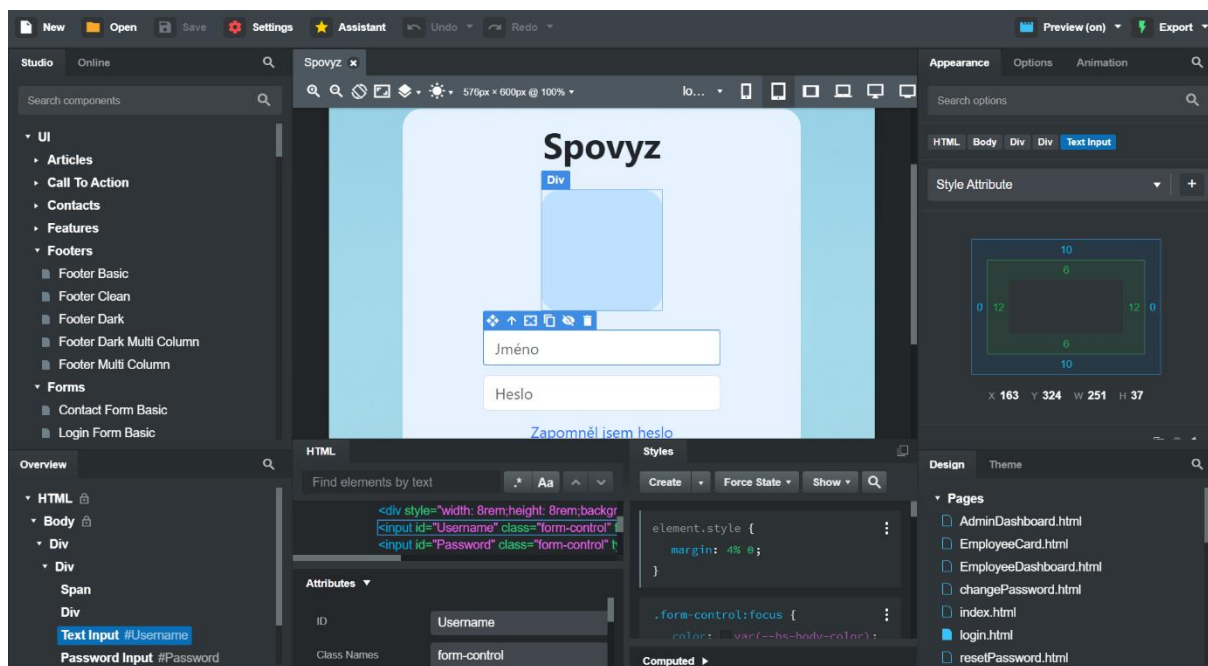


Obrázek 4: Původní návrh diagramu databáze pomocí Draw.io

## 2.2 Produktizace

Kódy do této části dokumentace jsem formátoval pomocí aplikace Syntax Highlighter. (23)

### 2.2.1 Vývoj v Bootstrap Studio



Obrázek 5: Ukazka aplikace Bootstrap Studio při vývoji login.html page

### 2.2.2 Generování hash pro heslo

```
1. byte[] salt = RandomNumberGenerator.GetBytes(128 / 8);
2. string hashedPassword =
   Convert.ToBase64String(KeyDerivation.Pbkdf2(
3.     password: password!,
4.     salt: salt,
5.     prf: KeyDerivationPrf.HMACSHA256,
6.     iterationCount: 100000,
7.     numBytesRequested: 256 / 8));
```

Kód 1: Funkčnost generování hash hesla

Toto je kód, který využívá C# knihovnu `Microsoft.AspNetCore.Cryptography.KeyDerivation` pro generování hash hesla.

### 2.2.3 Cors Policy

```
1. builder.Services.AddCors(options =>
2. {
3.     options.AddPolicy("CorsPolicy",
4.         builder => builder
5.             .WithOrigins("http://127.0.0.1:8000")
6.             .AllowAnyMethod()
7.             .AllowAnyHeader());
8. });
```

Kód 2: Cors Policy

Tato část slouží pro omezení přístupu na API pouze z IP adresy 127.0.0.1.

### 2.2.4 Připojení na server pomocí JavaScript

```
1. fetch('https://localhost:7221/api/Auth/token?' +
2.     params.toString(), { method: 'POST' })
3.     .then(response => {
4.         if (response.status === 401) {
5.             alert('Nejste přihlášení. Přihlaste se
6.             prosím znovu');
7.             window.location.href = 'login.html';
8.         }
9.         else if (response.status === 403) {
10.            alert('K těmto datům nemáte přístup');
11.        }
12.        else if (!response.ok) {
13.            const data = response.text();
14.            if (data === 'Uživatel nenalezen')
15.            {
16.                alert('Nejste přihlášení.
17.                Přihlaste se prosím znovu');
18.                window.location.href =
19.                'login.html';
20.            }
21.            else
22.                alert(data);
23.        }
24.        else
25.            return response.text();
26.    })
```

Kód 3: JavaScript kód pro přístup k API

Příkazem `fetch` pošleme požadavek na API metodou `POST`. Poté, co nám přijde odpověď, převedeme ji na `text`. Data uložíme do `local storage`. Pokud nastane během přenosu chyba, zobrazíme ji do konzole.

## 2.2.5 Tabulka pomocí Microsoft Entity Frameworku

```
1. public class Project
2. {
3.     public uint Id { get; set; }
4.     public string Name { get; set; }
5.     public string? Description { get; set; }
6.     public Customer Customer { get; set; }
7.     public DateOnly? Dead_line { get; set; }
8.     public Enums.Status Status { get; set; }
9.     public ICollection<Project_tag> Project_tags { get; set; }
10. }
11.     public ICollection<Project_employee>
    Project_employees { get; set; }
11. }
```

*Kód 4: Tabulka Project implementovaná jako C# třída*

Tento kód reprezentuje tabulku Project v databázi (pro použití Microsoft Entity Frameworku).

## 2.2.6 Použití Microsoft Entity Frameworku

```
1. Employee? activeUser = await _context.Employees
2.     .Include(e => e.Company)
3.     .FirstOrDefaultAsync(e => e.Username == Username);
```

*Kód 5: Použití Microsoft Entity Frameworku pro přístup k datům*

Tímto kódem získáme z databáze pomocí Microsoft Entity Frameworku informace o uživateli, který poslal aktuální požadavek na API.

## 2.2.7 Původní implementace kontroleru

```
1. [HttpGet("GetName/{id}")]
2. [Authorize]
3. public IActionResult GetName(int id)
4. {
5.     string error = "e1";
6.     string username = "";
7.
8.     Employee activeUser = _context.Employees.Include(e =>
        e.Company).FirstOrDefault(e => e.Username ==
        User.Identity.Name.ToString());
9.     Employee[] employees = [... _context.Employees
10.         .Include(e => e.Company)
11.         .Where(e => e.Company.Id ==
            activeUser.Company.Id)
12.         .ToArray()];
13.     if (id < 0 || id >= employees.Length)
14.         return Ok(new { error });
15.     return Ok(new { username = employees[id].Username
16. });
17. }
```

Kód 6: Původní implementace kontroleru

Původně jsem všechnu funkčnost psal přímo v kontroleru. To se rychle stalo nepřehledné a mě došlo, že takto se to určitě nedělá. Pro ukázkou je v projektu část kódu `EmployeeController.cs` napsána původním kódem, aby bylo vidět, jak jsem začínal.

## 2.2.8 Použitá implementace kontroleru

```
1. [HttpGet("{ProjectId}")]
2. [Authorize(Roles = "Worker, Supervisor, Manager")]
3. public async Task<IActionResult> Get(uint ProjectId)
4. {
5.     string? UserName = User.Identity?.Name?.ToString();
6.     if (UserName == null)
7.         return NotFound();
8.
9.     (ProjectCardData? data, string? error) = await
        _projectService.GetProjectById(UserName, ProjectId);
10.    if (error != null)
11.        return NotFound(error);
12.
13.    return Ok(data);
14. }
```

Kód 7: Použitá implementace kontroleru

Poradil jsem se s umělou inteligencí Gemini AI (19) a dostal jsem odpověď, aby použil vrstvenou architekturu. V mém případě jsem použil architekturu `kontroller-servis-repositář`.

## 2.2.9 Implementace servisu

```
1. public async System.Threading.Tasks.Task<(ProjectCardData?,  
   string?)> GetProjectById(string Username, uint ProjectId)  
2. {  
3.     Employee? activeUser = await  
   _context.Employees.Include(e =>  
   e.Company).FirstOrDefaultAsync(e => e.Username == Username);  
4.     if (activeUser == null)  
5.         return (null, "Uživatel nenalezen");  
6.     Project? project = await  
   _projectRepository.GetProjectById(ProjectId, activeUser.Id);  
7.     if (project == null)  
8.         return (null, "Projekt nenalezen");  
9.     string[]? tagNames = await  
   _tagRepository.GetTagNamesByProject(ProjectId);  
10.    uint[]? employeesIds = await  
   _employeeRepository.GetEmployeesIdsByProjectId(project.Id);  
11.    NameBasic[]? taskNames = await  
   _taskRepository.GetTaskNames(project, activeUser.Id);  
12.    uint? remains = null;  
13.    if (project.Dead_line != null)  
14.        remains =  
   (uint)((DateOnly)project.Dead_line).ToDateTime(TimeOnly.Min  
   Value) - DateTime.Now).Days;  
15.    ProjectCardData data = new ProjectCardData()  
16.    {  
17.        Name = project.Name,  
18.        Description = project.Description,  
19.        Customer = project.Customer.Id,  
20.        Status = (uint)project.Status,  
21.        Deadline = project.Dead_line,  
22.        Remains = remains,  
23.        Tags = tagNames,  
24.        Employees = employeesIds,  
25.        Tasks = taskNames  
26.    };  
27.    return (data, null);  
28. }
```

Kód 8: Implementace servisu

Předchozí kód odkazuje na servis `_projectService`. Toto je taková mezivrstva mezi přístupem k API (kontroler) a přístupem k databázi (repositář). Zde kontroluji data a připravuji je na vložení či získání z databáze.

### 2.2.10 Implementace repositář

```
1. public async System.Threading.Tasks.Task<Project?>
   GetProjectById(uint ProjectId, uint activeUserId)
2. {
3.     return await _context.Project_employees
4.         .Include(pe => pe.Project)
5.         .ThenInclude(p => p.Customer)
6.         .Include(pe => pe.Employee)
7.         .Where(pe => pe.Project.Id == ProjectId &&
   pe.Employee.Id == activeUserId)
8.         .Select(pe => pe.Project)
9.         .FirstOrDefaultAsync();
10. }
```

*Kód 9: Implementace repositáře*

Repositář je jediná část serveru, která má přímý přístup k databázi. Do repositáře se dostávají již ověřená a zvalidovaná data.

### 2.2.11 Práva přístupu ke kontroleru

```
1. [HttpGet("{ProjectId}")]
2. [Authorize(Roles = "Worker, Supervisor, Manager")]
3. public async Task<IActionResult> Get(uint ProjectId)
```

*Kód 10: Práva přístupu ke kontroleru*

Ve výchozím nastavení mají přístup ke kontrolerům úplně všichni. Já nastavil, aby přístup byl umožněn pouze těm uživatelům, kteří se prokážou platným tokenem. Nakonec jsem nastavil, aby kromě toho, se ke konkrétnímu kontroleru dostal pouze uživatel s patřičným oprávněním.

### 2.2.12 Validace dat

```
1. private static bool Email(string email)
2. {
3.     string pattern = @"^[^@\s]+@[^@\s]+\.[^@\s]+$";
4.     if (Regex.IsMatch(email, pattern))
5.         return true;
6.     else
7.         return false;
8. }
```

*Kód 11: Validace dat*

Protože na API sice mohou přijít požadavky se správným datovým typem přijímaných dat, neznamená to, že tato data jsou v pořádku. Proto jsem vytvořil script, který kontroluje data před manipulací s databází.

### 2.2.13 Logo

Tvorba loga mi zabrala spoustu času a práce. Nikdy jsem nepracoval se žádným softwarem pro tvorbu grafiky a téměř všechny obrázky do projektu jsou vygenerované pomocí umělé inteligence Gemini AI. (19) Musel jsem se tedy naučit webový editor Photopea (23) úplně od základů.

Chtěl jsem si ale vyzkoušet práci na grafice, a tak celé logo je čistě má práce.



Obrázek 6: Logo

## 2.3 Popis pro uživatele

Uživatel dostane od administrátora uživatelské jméno a heslo, kterým se následně přihlásí. Aplikace je velmi přehledná a intuitivní.



## Závěr

---

Se svou prací jsem spokojený. Během vývoje tohoto projektu jsem se naučil spoustu nových dovedností a zkusil jsem si vývoj jak webové stránky pomocí Bootstrap Studio, tak i vývoj API ve Visual Studiu.

Nejvíce času mi zabralo tvoření designu. Postupoval jsem dle návrhu a vývojová aplikace Bootstrap Studio mi při tvorbě velice pomohla. I přesto jsem měl celkem velké problémy při pozicování vizuálních a interaktivních prvků na stránce. Všechny prvky jsem měl připravené, ale nedařilo se mi je usadit na správné místo.

Další problém byla responzivita webové stránky. Bootstrap je framework, který obsahuje spoustu předpřipravených responzivních prvků. Nicméně mnoho z nich mi pro vývoj webové aplikace nevyhovovalo nebo se prvky na velikostně menších obrazovkách chovaly nestandardně. Všechny tyto problémy jsem po usilovné práci zvládnul vyřešit.

Použití Microsoft Entity Frameworku v kombinaci s databází PostgreSQL proběhlo zcela podle plánu. Dokumentace k tomuto frameworku je velice přehledná a snadno využitelná.

Vývoj samotného API nebyl příliš složitý, ale obsahoval spoustu částí důležitých na pochopení funkčnosti. Veškeré, mě neznámé funkčnosti, jsem se postupně naučil a jsem moc rád za nové zkušenosti.

Ze začátku vývoje jsem měl drobné problémy s připojením se na své API z webové stránky tak, abych zajistil co největší zabezpečení API. To se později ukázalo jako snadno nastavitelné a poté jsem mohl přístup ke svému API plně využívat.

Rozvržení plánu do zadávacího listu mi mírně nevyhovovalo. Při zadávání jsem neměl tušení, kolik času některé implementace funkcí zaberou, a proto jsem se občas dostal do časové tísně.

Také jsem se naučil, že je dobré si některé funkčnosti dobře navrhnout. A i přesto, že můžeme udělat sebelepší plán, vždy se může stát, že postupem vývoje projektu se narazí na problémy, se kterými návrh nepočítal, a občas je potřeba část návrhu předělat. Žádný můj původní návrh se neobešel beze změny.

## Použitá literatura

---

1. Microsoft. Co je DevOps? Vysvětlení DevOps | Microsoft Azure. *Microsoft Learn*. [Online] 2. Květen 1991. [Citace: 31. Říjen 2024.] <https://azure.microsoft.com/cs-cz/resources/cloud-computing-dictionary/what-is-devops>.
2. Šmarda, Jaroslav. Mzdový a personální informační systém. *Vema*. [Online] 31. Červenec 1996. [Citace: 20. Říjen 2024.] [https://www.vema.cz/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=Brand%20\(hlavn%C3%AD\)&utm\\_id=675492716&gad\\_source=1&gclid=EAlalQobChMIjbTBp9vwigMVdqqDBx1G2QB-EAAYASAAEgJP2PD\\_BwE](https://www.vema.cz/?utm_source=google&utm_medium=cpc&utm_campaign=Brand%20(hlavn%C3%AD)&utm_id=675492716&gad_source=1&gclid=EAlalQobChMIjbTBp9vwigMVdqqDBx1G2QB-EAAYASAAEgJP2PD_BwE).
3. Calm. Calm - The #1 App for Meditation and Sleep. *Calm*. [Online] 11. Červenec 1995. [Citace: 20. Říjen 2024.] <https://www.calm.com/>.
4. Ducháčková, Daniela. Psychologie barev v marketingu: Význam barev? *Orange Academy*. [Online] 4. Leden 2021. [Citace: 20. Říjen 2024.] <https://orangeacademy.cz/clanky/psychologie-barev-v-marketingu/>.
5. rkp. Vhodné pracovní prostředí jako klíč k produktivitě a zdraví zaměstnanců | Hospodářské noviny (HN.cz). *Hospodářské Noviny*. [Online] 26. Květen 2017. [Citace: 20. Říjen 2024.] <https://hn.cz/c1-65742430-vhodne-pracovni-prostredi-jako-klic-k-produktivite-a-zdravi-zamestnancu>.
6. Ovlivňuje vzhled pracoviště produktivitu? *Leitz*. [Online] 27. Září 2023. [Citace: 20. Říjen 2024.] <https://www.leitz.com/cs-cz/blog/leitz/workleitzbalance/do-the-aesthetics-of-your-workplace-affect-productivity/>.
7. Ipsos. 4 z 10 mladých Čechů v posledním roce absentovali v práci kvůli stresu | Ipsos. *Ipsos*. [Online] 10. Říjen 2024. [Citace: 20. Říjen 2024.] <https://www.ipsos.com/cs-cz/4-z-10-mladych-cechu-v-poslednim-roce-absentovali-v-praci-kvuli-stresu>.
8. 1. lékařská fakulta Univerzity Karlovy. Jak je na tom psychika mladých dospělých? Vědci ji v unikátním výzkumu studovali deset let. - 1. lékařská fakulta Univerzity Karlovy. *1. lékařská fakulta Univerzity Karlovy*. [Online] 21. Květen 2024. [Citace: 20. Říjen 2024.] <https://www.lf1.cuni.cz/jak-je-na-tom-psychika-mladych-dospelych-vedci-ji-v-unikatnim-vyzkumu-studovali-deset-let>.
9. Kliková, Hana. Pracovní podmínky a jejich vliv na výkon. *Digitální knihovna Západočeské univerzity v Plzni*. [Online] Březen 2017. [Citace: 20. Říjen 2024.] [https://otik.uk.zcu.cz/bitstream/11025/29010/1/BP\\_Pracovni\\_podminky\\_a\\_jejich\\_vliv\\_na\\_vykon\\_za\\_mestnance.pdf](https://otik.uk.zcu.cz/bitstream/11025/29010/1/BP_Pracovni_podminky_a_jejich_vliv_na_vykon_za_mestnance.pdf).
10. Kmoníčková, Šárka. VLIV PRACOVNÍHO PROSTŘEDÍ NA VÝKONNOST. *Theses.cz – Vysokoškolské kvalifikační práce*. [Online] 2009. [Citace: 20. Říjen 2024.] <https://theses.cz/id/xgnas2/66703-334966566.pdf>.
11. Microsoft. Dokumentace jazyka C# – začínáme, kurzy, reference. | Microsoft Learn. *Microsoft Learn*. [Online] 2. Květen 1991. [Citace: 31. 10 2024.] <https://learn.microsoft.com/cs-cz/dotnet/csharp/tour-of-csharp/>.
12. Bootstrap studio and their contributors. Get started with Bootstrap · Bootstrap v5.3. *Get Bootstrap*. [Online] 1. Leden 2012. [Citace: 31. Říjen 2024.] <https://getbootstrap.com/docs/5.3/getting-started/introduction/>.

13. steveoh. npgsql/README.md at main · npgsql/npgsql. *Github*. [Online] 15. Březen 2023. [Citace: 31. Říjen 2024.] <https://github.com/npgsql/npgsql/blob/main/README.md>.
14. Microsoft. Entity Framework documentation hub | Microsoft Learn. *Microsoft Learn*. [Online] 2. Květen 1991. [Citace: 31. Říjen 2024.] <https://learn.microsoft.com/en-us/ef/>.
15. Bootstrap Studio Docs | Bootstrap Studio. *Bootstrap Studio*. [Online] 11. Červenec 2015. [Citace: 20. Říjen 2024.] <https://bootstrapstudio.io/docs/>.
16. Microsoft. Dokumentace sady Visual Studio | Microsoft Learn. *Microsoft Learn*. [Online] 2. Květen 1991. [Citace: 31. Říjen 2024.] <https://learn.microsoft.com/cs-cz/visualstudio/windows/?view=vs-2022&preserve-view=true>.
17. —. Documentation for Visual Studio Code. *Visual Studio Code*. [Online] 18. Březen 1997. [Citace: 31. Říjen 2024.] <https://code.visualstudio.com/docs>.
18. AI Image Upscaler | Upscale & enhance photos. *AI Nero*. [Online] 4. Listopad 1997. [Citace: 1. Leden 2025.] [https://ai.nero.com/image-upscaler/?gad\\_source=1&gclid=EAlaIqobChMI0cWlg5\\_ZigMVL42DBx1HzCeEAAAYASAAEgIkHfD\\_BwE](https://ai.nero.com/image-upscaler/?gad_source=1&gclid=EAlaIqobChMI0cWlg5_ZigMVL42DBx1HzCeEAAAYASAAEgIkHfD_BwE).
19. Google. Gemini. *Gemini*. [Online] 15. Září 1997. [Citace: 31. Říjen 2024.] <https://gemini.google.com/u/1/app?hl=cs>.
20. Figma. *Figma*. [Online] 10. Duben 1999. [Citace: 31. Říjen 2024.] <https://www.figma.com/>.
21. MyScript. Nebo. *Nebo*. [Online] 8. Květen 2018. [Citace: 31. Říjen 2024.] <https://www.nebo.app/>.
22. Draw.io. *Draw.io*. [Online] 17. Březen 2010. [Citace: 31. Říjen 2024.] <https://app.diagrams.net/>.
23. RIPE Network Coordination Centre (RIPE). Photopea | Online Photo Editor. *Photopea*. [Online] 1. 5 1993. [Citace: 29. 5 2025.] <https://www.photopea.com/>.
24. Yurevick, Shevchuk Anton. Syntax Highlighter. *Syntax Highlighter*. [Online] 10. Březen 2007. [Citace: 12. Leden 2025.] <https://highlight.hohli.com/>.
25. HonzikVacik. HonzikVacik/Spovyz. *Github*. [Online] 31. Říjen 2024. [Citace: 31. Říjen 2024.] <https://github.com/HonzikVacik/Spovyz>.
26. Microsoft. ASP.NET documentation | Microsoft Learn. *Microsoft Learn*. [Online] 2. Květen 1991. [Citace: 31. Říjen 2024.] <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-9.0>.

## Seznam obrázků

---

Obrázek 1: Část návrhu propojení všech stránek webu v Nebo .....	15
Obrázek 2: Část návrhu designu ve Figmě .....	16
Obrázek 3: Část návrhu databáze v Nebo .....	16
Obrázek 4: Původní návrh diagramu databáze pomocí Draw.io .....	17
Obrázek 5: Ukazka aplikace Bootstrap Studio při vývoji login.html page .....	18
Obrázek 6: Logo .....	24

## Seznam kódů

---

Kód 1: Funkčnost generování hash hesla .....	18
Kód 2: Cors Policy .....	19
Kód 3: JavaScript kód pro přístup k API.....	19
Kód 4: Tabulka Project implementovaná jako C# třída .....	20
Kód 5: Použití Microsoft Entity Frameworku pro přístup k datům .....	20
Kód 6: Původní implementace kontroleru .....	21
Kód 7: Použitá implementace kontroleru .....	21
Kód 8: Implementace servisu .....	22
Kód 9: Implementace repositáře.....	23
Kód 10: Práva přístupu ke kontroleru .....	23
Kód 11: Validace dat.....	23

## Obsah média

---

- Databáze.sql
- Dokumentace
  - Dokumentace.docx
  - Dokumentace.pdf
- Kopie Gitu
- Návrhy
- Obrázky
- Odkaz na GitHub
- Prezentace.pptx
- Projekt
  - Databáze.sql
  - Frontend
  - Server