

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Webová aplikace s texty Bible



2025

Vedoucí práce:  
Mgr. Tomáš Urbanec, Ph.D.

Patrik Prinz

Studijní program: Informatika,  
Specializace: Programování a vývoj  
software

## **Bibliografické údaje**

Autor: Patrik Prinz  
Název práce: Webová aplikace s texty Bible  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2025  
Studijní program: Informatika, Specializace: Programování a vývoj software  
Vedoucí práce: Mgr. Tomáš Urbanec, Ph.D.  
Počet stran: 10  
Přílohy: elektronická data v úložišti katedry informatiky  
Jazyk práce: český

## **Bibliographic info**

Author: Patrik Prinz  
Title: Web application with Bible texts  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2025  
Study program: Computer Science, Specialization: Programming and Software Development  
Supervisor: Mgr. Tomáš Urbanec, Ph.D.  
Page count: 10  
Supplements: electronic data in the storage of department of computer science  
Thesis language: Czech

## Anotace

*Ukázkový text závěrečné práce na Katedře informatiky Přírodovědecké fakulty Univerzity Palackého v Olomouci, který je zároveň dokumentací stylu pro text práce v  $\text{\LaTeX}$ . Zdrojový text v  $\text{\LaTeX}$  je doporučeno použít jako šablonu pro text skutečné závěrečné práce studenta.*

## Synopsis

*Sample text of thesis at the Department of Computer Science, Faculty of Science, Palacký University Olomouc and, at the same time, documentation of the  $\text{\LaTeX}$  style for the text. The source text in  $\text{\LaTeX}$  is recommended to be used as a template for real student's thesis text.*

**Klíčová slova:** webová aplikace; Bible; PWA; TypeScript; DevOps; CI/CD

**Keywords:** web application; Bible; PWA; TypeScript; DevOps; CI/CD

Děkuji, děkuji, děkuji.

*Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>Podobné projekty</b>	<b>7</b>
2.1	.....	8
<b>3</b>	<b>Struktura projektu</b>	<b>8</b>
3.1	Balíčky .....	8
3.1.1	Kořenový balíček .....	8
3.2	Nastavení projektu .....	8
3.3	Repozitář a CI/CD pipeline .....	8
3.4	Styl kódu .....	8
3.5	Backend .....	9
3.5.1	Struktura modulu .....	9
3.6	Frontend .....	9
3.7	Databáze .....	9
<b>4</b>	<b>Použité technologie</b>	<b>9</b>
4.1	TypeScript .....	9
4.2	Express .....	9
4.3	Elastic search .....	9
4.4	Docker .....	9
4.5	Github actions .....	9
	<b>Seznam zkratk</b>	<b>10</b>

## Seznam tabulek

# 1 Úvod

Cílem této práce je Vytvořit aplikaci pro použití v Řeckokatolické farnosti v Olomouci. Farnost je v současnosti vícenárodnostní a v posledních letech se značně rozrůstá. nejpočetněji jsou zde zastoupeni farníci z Ukrajiny. Texty Bible jsou v katolické církvi velmi často využívané při liturgických obřadech, ale také při osobní modlitbě a rozjímání.

Pro Apoštolský Exarchát České Republiky, který je nadřazenou organizací Olomoucké farnosti existuje kalendář, který obsahuje i biblické texty pro každý den v roce. Biblické texty v církevních obřadech jsou často vázány ke svátkům, které jsou buď pevně svázány s datem, nebo jsou pohyblivé a připadají každým rokem na jiné datum. Mnoho textů je také svázáno s konkrétním obřadem, nebo událostí (např. křest, svěcení domu...).

Texty z knihy žalmů jsou často často používány jako samostatné modlitby. Východní církve žalmy rozděluje do 20 skupin zvaných kaftizmy, které obsahují většinou okolo 4-6 žalmů. Ke kaftizmům jsou přidruženy další texty, které se modlí zároveň s nimi. Farnost má vlastní společenství, které se modlí modlitby žalmů a funguje přibližně od roku 2024. Každý člen společenství se modlí jednu z kaftizem během týdne. Kaftizmy se mezi členy společenství pravidelně střídají.

Nenašel jsem dosud žádný projekt, který by poskytoval možnosti práce s biblickými texty v tomto rozsahu. Proto chci vytvořit aplikaci, která by sdružovala by texty více jazyků s možností vlastní správy a přidávání dalších překladů a textů pro vlastní četbu a studium. S tímto bude spojena i možnost udržování vlastního kalendáře který bude obsahovat zejména úryvky na každý den. Také by zde měla být obsažena možnost vytvářet si a ukládat vlastní úryvky. Proces rozdělení kaftizem v rámci společenství žaltáře plánuji zautomatizovat a zároveň poskytnout na stejném místě i texty samotné i spřidruženými modlitbami spolu s možností označit, že se člen již kaftizmu pomodlil.

Dále bych chtěl, aby platforma byla do budoucna dále rozšiřitelná a bylo možné přidávat další funkcionality a aby mohl projekt i v budoucnu reflektovat požadavky společenství, které jej bude využívat.

Dále se chci v této práci zaměřit na průzkum a využití konceptů z oblasti DevOps, které usnadní vývoj samotný, urychlí integraci a nasazení aplikace. Na projektu pracuji sám, proto je pro mne výhodné především zautomatizovat proces ověření a nasazení software a sledování jeho stavu. I za cenu větší náročnosti přípravy a počáteční konfigurace je pro mne výhodné získat značně větší spolehlivost a rychlost nasazení a také do budoucna usnadní řadu problémů s udržováním potřebné infrastruktury. Také doufám, že nabyté znalosti z této oblasti využiji i v budoucnu ve svém profesním životě.

## 2 Podobné projekty

V rámci této práce jsem hledal a procházel i další software zaměřený na práci s texty Bible i dalšími liturgickými texty. Aplikací s tímto zaměřením existuje

celá řada. Většina je však určena k jiným účelům a především pro osobní použití, ne pro použití ve farnosti samotné, případně v jiném společenství. Podobné projekty bývají zaměřeny na samotné čtení a studium biblických textů. Většina z projektů má také uzavřený zdrojový kód a proto je nelze upravit a přizpůsobit pro specifické potřeby, nebo je již zastaralá a dlouhodobě neudržovaná. Mnou vyvíjená aplikace má řadu velmi specifických požadavků Proto má jen malý průnik s řadou

## 2.1

# 3 Struktura projektu

Projekt je strukturován do monorepozitáře, který obsahuje několik balíčků (packages), do kterých je funkcionalita rozdělena. Kořenový balíček monorepozitáře obsahuje společnou konfiguraci nástrojů a společné balíčky.

Pro správu jsem použil balíčkovací nástroj pnpm pro node.js, který je alternativou k npm. Pro projekty s větším množstvím závislostí poskytuje značně lepší výkon a také možnost používat workspaces. Workspaces umožňují mít v rámci projektu více balíčků s vlastní konfigurací a vlastními závislostmi.

## 3.1 Balíčky

### 3.1.1 Kořenový balíček

Kořenový balíček má na starosti uchovávání konfigurace

## 3.2 Nastavení projektu

Popis Základní struktury projektu, konfigurace a popis složek a souborů v kořenovém balíčku.

## 3.3 Repozitář a CI/CD pipeline

Pro verzování projektu jsem použil GIT a jako vzdálené úložiště GitHub. Nástroj GIT je velmi populární a v současnosti má velmi málo použitelných alternativ. GitHub jsem zvolil také pro jeho rozšířenost a Řadu možností, které poskytuje. mezi ně patří především služba GitHub Actions, kterou jsem použil pro vytvoření CI/CD pipeline.

GitHub Actions je flexibilní nástroj, který umožňuje spouštět akce, jako reakci na události v repozitáři.

## 3.4 Styl kódu

Popis konvencí



## 3.5 Backend

Sdílené soubory

Aplikace na backendu má modulární monolitickou architekturu, která zajišťuje rozdělení funkcionalit do skupin podle jejich domény a tím usnadňuje i kontrolu závislostí a umožňuje snadnější rozšiřitelnost.

Jádrem aplikace je sada globálně dostupných nástrojů, které implementují funkcionality, které jsou nezávislé na doméně a používají se na více místech napříč aplikací.

Patří sem například middleware, který zajišťuje validaci dat, ošetření chyb a logování. Dále také sdílené knihovny, jako například databázový adaptér.

Moduly

### 3.5.1 Struktura modulu

Moduly sdružují funkcionalitu, která je spojena na úrovni domény projektu. Každý modul obsahuje:

- **router:** mapuje HTTP požadavky na příslušné metody controlleru
- **controller:** Validuje a zpracovává data a požadavky. Zajišťuje komunikaci routerem a spouští service
- **service:** zajišťuje získání a zpracování dat na základě požadavku. Výsledek vrací controlleru. Zajišťuje komunikaci mezi jednotlivými použitými závislostmi. Uchovává závislosti
- **repository:** zajišťuje komunikaci s databází prostřednictvím adaptéru a poskytuje metody specifické pro danou doménu. Provádí překlad mezi doménovými typy a typy pro elasticsearch klienta. Zajišťuje kontrolu a validaci dat. Obstarává konzistentnost dat v databázi.

## 3.6 Frontend

## 3.7 Databáze

# 4 Použité technologie

## 4.1 TypeScript

## 4.2 Express

## 4.3 Elastic search

## 4.4 Docker

## 4.5 Github actions

