

Pseudodokumentace MP

V tomto dokumentu jsou rozepsané informace týkající se mé MP. Nejsou však ve formě, v jaké budou ve finální dokumentaci projektu.

Výběr frameworku

Jelikož je poslední dobou velmi populární jazyk Rust, zejména pro svůj striktní typový systém, který pomáhá při kompilaci zachytit chyby, které jsou v jiných jazycích často detekovány až při běhu programu (což už je pozdě), zvažoval jsem jej použít i pro tento projekt.

Nejvíce mě po krátkém bádání zaujaly projekty **Leptos** a **Tauri**.

1. [Tauri](#) využívá FFI (foreign function interface) k volání Rust funkcí z JS a naopak. Web by tedy využíval trojici jazyků HTML, CSS a JS, přičemž JS (který v oblibě nemám) by fungoval jen jako takový „most“ pro předávání dat Rust backendu a zpět JS.
2. [Leptos](#) umožňuje pomocí procedurálních maker psát HTML kód přímo mezi řádky Rust kódu. Zdrojový kód aplikace by tak byl vysoce expresivní. Navíc je možné pomocí frameworku Leptos zkompilevat projekt do aplikace, která nevyžaduje žádné externí nainstalované nástroje.

Postup pro nasazení výsledné aplikace na server by mohl vypadat nějak takto:

1. Nahrání aplikace na server (např. do /var/www/moje_rust_app) pomocí SFTP.
2. Přidělení nezbytných práv souboru.
3. Vytvoření souboru služby (např. /etc/systemd/system/moje_rust_app.service) pro správu aplikace.

Obsah souboru (jen příklad):

Spuštění příkazů:

sudo systemctl enable moje_rust_app, sudo systemctl start moje_rust_app

```
[Unit]
Description=Moje Rust Web Aplikace
After=network.target

[Service]
User=www-data
Group=www-data
WorkingDirectory=/var/www/moje_rust_app
ExecStart=/var/www/moje_rust_app/muj_web_server
Restart=always
RestartSec=5s
StandardOutput=syslog
StandardError=syslog

[Install]
WantedBy=multi-user.target
```

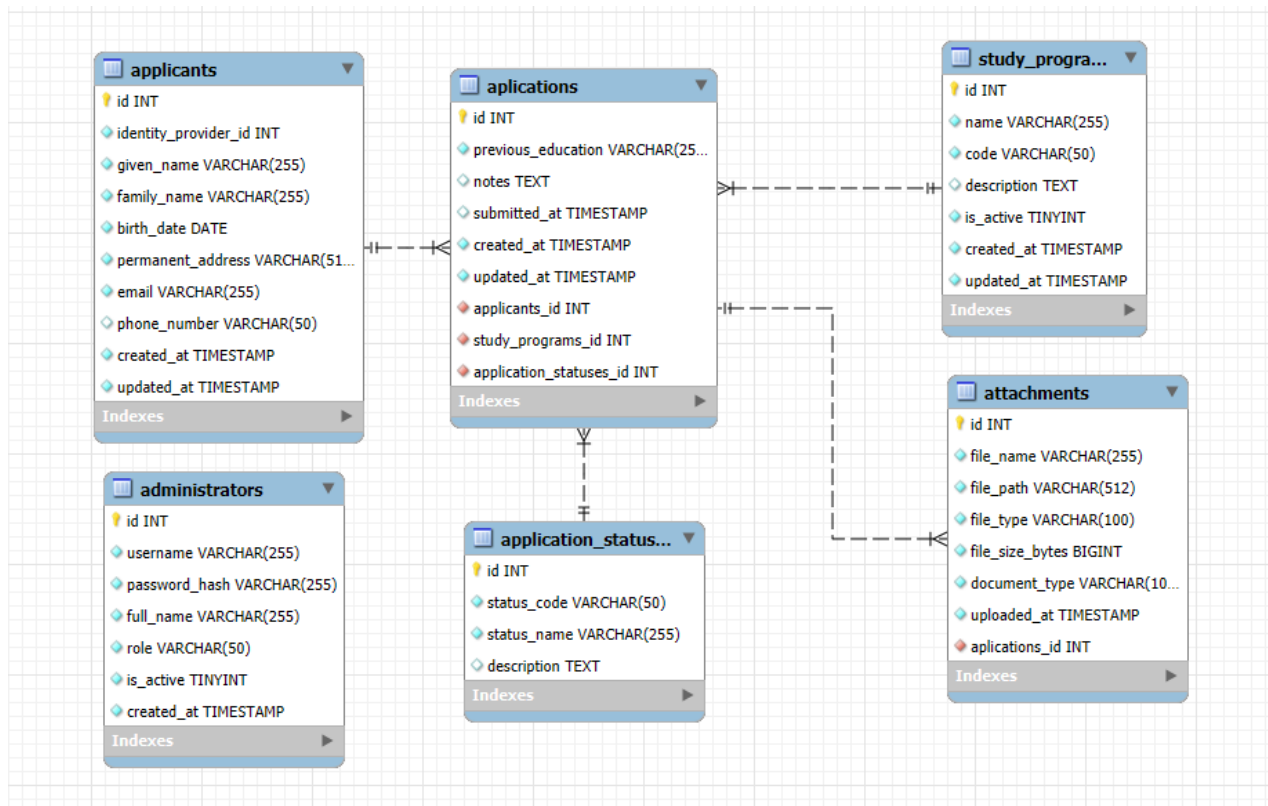
Ačkoliv se obě možnosti zdají být velmi atraktivní, zjistil jsem, že VPS, který by webová aplikace vytvořená pomocí některého z Rust frameworků vyžadovala je mnohem dražší než běžný webhosting (webhosting: ~73 Kč/měsíc vs VPS: ~300 Kč/měsíc).

Začal jsem proto bádát po PHP frameworku, který bych pro tento projekt využil. Nakonec jsem se však (prozatím) stejně rozhodl pro Laravel, se kterým mám již spoustu (dobrých) zkušeností (viz mnou vytvořené weby: arbo-kytka.com, strongbull.cz, navrhy.altech.cz). Navíc mi Laravel ve srovnání s ostatními PHP frameworky připadá nejstabilnější a nabízí řadu nástrojů pro zjednodušení práce (dobrá dokumentace, rozšíření do editoru, ...).

Po diskuzi s panem zástupcem jsem zjistil potřebné informace k volbě té správné technologie:

1. Webová aplikace bude nasazena na stejný hosting, jako je web školy (www.active24.cz).
2. Operační systém je linux.
3. Verze nainstalovaného interpretu pro PHP je 8.4 (nejnovější).
4. Dostupný webový server je Apache 2.4.
5. Dostupná databáze je MySQL (MariaDB).
6. Webové aplikaci bude vytvořena vlastní subdoména, tzn., že tento projekt nebude zasahovat do souboru aktuálního webu školy.

Návrh databáze (struktura se v průběhu projektu bude pravděpodobně několikrát měnit)



- **applicants (Uchazeči)** - Uchovává seznam všech lidí, kteří se přihlásili. Obsahuje jejich státem ověřené údaje (jméno, příjmení, datum narození) a kontakty, které sami doplnili (e-mail, telefon).
- **study_programs (Studijní obory)** - Je to číselník všech studijních oborů, které škola nabízí (bude jen jeden, ale pro případ, že by se aplikace měla v budoucno rozšířit). Administrátoři zde mohou přidávat nebo upravovat obory, na které se dá podat přihláška.
- **application_statuses (Stavy přihlášky)** - Další číselník, tentokrát pro stavy, ve kterých se přihláška může nacházet (např. „Nová“, „Přijata“, „Zamítnuta“). Pomáhá to sledovat, co se s přihláškou děje.
- **applications (Přihlášky)** - Nejdůležitější tabulka. Představuje jednu konkrétní přihlášku. Propojuje uchazeče (applicants) se studijním oborem (study_programs) a ukazuje její aktuální stav (application_statuses).
- **attachments (Přílohy)** - Zde jsou informace o souborech, které uchazeč nahrál ke své přihlášce (např. maturitní vysvědčení). Neukládá samotný soubor, jen jeho název a cestu, kde je na serveru uložený.
- **administrators (Administrátoři)** - Seznam uživatelských účtů pro pracovníky školy (studijní oddělení), aby se mohli přihlásit do systému a spravovat doručené přihlášky.

Návrh propojení webové aplikace s aktuálním webem školy (umístění a design se dost možná změní)

O

A

U

H

STŘEDNÍ ŠKOLA

✓

VYŠŠÍ ODBORNÁ ŠKOLA

JAZYKOVÁ ŠKOLA

UNIVERZITA TŘETÍHO VĚKU

DALŠÍ VZDĚLÁVÁNÍ

21.11.2025 - 22.11.2025

Dny otevřených dveří

→

18.09.2025 - 30.09.2025

Výsledky 3. kola přijímacího řízení na VOŠ

→

01.09.2025

Pár slov ředitele školy k zahájení nového školního roku 2025/2026

→

→ více

Informace o oborech

Vyšší odborná škola

Ekonomicko-právní činnost
(68-41-N/03)

Hledáš pomaturitní studium, které Tě připraví do praxe?

→ Pojď studovat VOŠku

Elektronická přihláška

Ověřeno státem
(BankID, eDoklady)

Poděj si přihlášku rychle, moderně a bezpečně z pohodlí domova.

→ Podat e-přihlášku

Po kliknutí na tlačítko „Podat e-přihlášku“ by byl návštěvník stránky přesměrován do webové aplikace. Například <https://prihlasky.oauh.cz>.

Založení GitHub repozitáře

Tento soubor lze najít také v repozitáři projektu této MP: <https://github.com/Eightzi4/OAUH-MP-prihlasky-VOS>

Registrace do aplikace eDoklady

Dne 20.10. jsem se pomocí účtu mé banky (Fio banka) přihlásil do aplikace, abych si sám prošel procesem, kterým projde každý zájemce o studium na VOŠ při použití této aplikace.

Načítání aplikace včetně načítání po přihlášení je pomalé, cca 22 vteřin (testováno na Poco F7 Pro).

Přiložil bych snímek obrazovky, jenže aplikace nepovoluje jej pořídít.

Zdroje:

<https://github.com/flosse/rust-web-framework-comparison>

<https://leptos.dev/>

<https://v2.tauri.app/>

<https://laravel.com/>

[https://www.reddit.com/r/PHP/comments/y9m3cu/what are the popular php frameworks you are using/](https://www.reddit.com/r/PHP/comments/y9m3cu/what_are_the_popular_php_frameworks_you_are_using/)

<https://www.active24.cz/>

<https://www.oauh.cz/>