

Registrační systém pro akci Plejády

Martin Rosenberg, O8.A.

Předmět: Informatika

Vedoucí: Vojtěch Horký

Školní rok: 2022/2023



Gymnázium, Praha 6, Nad Alejí 1952
Nad Alejí 1952/5, 162 00 Praha 6

Zadání maturitní práce z informatiky

Jméno žáka: Martin Rosenberg
Název práce: Registrační systém pro akci Plejády
Termín odevzdání: 14. dubna 2023
Jméno vedoucího: Vojtěch Horký

Rozpis podmínek na jednotlivé části práce

Obecné podmínky odevzdání popisuje příloha společná pro všechny maturanty.

Obsah práce

Naprogramujte aplikaci pro správu školní akce Plejády.

Funkce

Cílem aplikace je umožnit jednoduché elektronické přihlašování na jednotlivé přednášky Plejád. Systém musí být použitelný pro více let, tj. musí nabídnout funkce pro vytvoření nového ročníku a archivaci (popř. smazání) starších akcí.

Administrační rozhraní

Aplikace nabídne administrační rozhraní pro správu přednášek (včetně možnosti archivace celé akce). U každé přednášky je pak možné vyplnit kromě názvu a místa také kapacitu místnosti.

Přihlašování účastníků k systému

Předpokládá se napojení na existující systém (např. Google účty) tak, aby uživatelé mohli využít funkci SSO a správci pokud možno nemuseli vytvářet účty ručně.

Přihlašování k přednáškám

Aplikace nabídne uživatelům přihlašování k přednáškám a sama zajistí, že se půjde přihlásit pouze na přednášku s dosud volnou kapacitou.

Důraz je na zpracování v reálném čase, kdy uživatel ihned vidí volné přednášky a počet volných míst.

Přehledové funkce

Aplikace nabídne uživateli přehled zapsaných přednášek (alternativně též skrz rozhraní jiné aplikace jako je třeba import do kalendáře apod.).

Specifické pokyny pro vypracování

Řešitel využije existující knihovny pro vytvoření ORM vrstvy a pro funkce přihlašování (varianta SSO).

Toto zadání práce schválil dle příslušné maturitní vyhlášky a žákovi zadal ředitel školy.

V Praze dne 9. prosince 2022

.....
podpis ředitele a razítko

Prohlášení

Prohlašuji, že jsem svou maturitní práci na téma *Registrační systém pro akci Plejády* vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu zdrojů.

V Praze dne 14. 4. 2023

Anotace

Projekt si klade za cíl zjednotit a zoptimalizovat přihlašování studentů během studenty organizovaného přednáškového dne Plejády. Pro účastníky je připravený intuitivní a jednoduchý portál, ve kterém se mohou na jednotlivé přednášky přihlásit jediným kliknutím a v reálném čase sledovat obsazenost. Pro organizátory je tu připravené administrátorské rozhraní, ve kterém mohou celý přednáškový den spravovat a nastavit si vše dle své potřeby.

Klíčová slova

přihlašovací systém, real-time dashboard, webová aplikace, Elixir, fault-tolerant system

Annotation

The project's primary goal is to make the users' experience with signing up for talks during the school event Plejády as comfortable as possible. Therefore, the attendees are presented with an intuitive and easy-to-use single-page application, where they can sign up for talks with a single click and see the current occupancy of individual talks in real-time. Furthermore, event administrators can easily manage the whole event through the admin panel.

Keywords

event management system, real-time dashboard, web application, Elixir, fault-tolerant system

Obsah

1. Úvod	6
2. Motivace	6
3. Hlavní funkce	6
3.1. Jednotné přihlašování do systému	6
3.2. Přihlašování k přednáškám	6
3.3. Administrátorské rozhraní	7
4. Technologie	7
5. Závěr	7
6. Odkazy	8
7. Zdroje	8
7.1. Použité knihovny	8
7.2. Další použitá dokumentace	8

1. Úvod

V rámci své maturitní práce jsem se věnoval tvorbě přihlašovacího systému pro školní přednáškový den organizovaný studenty zvaný Plejády. Jelikož má tato akce na naší škole již mnohaletou tradici, je pochopitelné, že se s časem vyvíjel i systém, jakým se studenti mohli na jednotlivé přednášky přihlásit. V úplných začátcích se používaly papíry rozvěšené na nástěnce, kam se studenti v jeden moment nahrnuli a snažili se dostat na přednášky, na které chtějí jít. Poté se k přihlašování chvíli používaly Formuláře Google, které ale neumožňovali sledovat, kolik lidí je kam přihlášeno a tak se stávalo, že se velké procento studentů nedostalo na přednášky, na které skutečně chtějí.

2. Motivace

Před začátkem každého ročníku se organizátoři museli shodnout na způsobu přihlašování účastníků. To ovšem zabíralo poměrně hodně času a naskytla se tak vhodná příležitost na vytvoření jednotného přihlašovacího systému, který mohl být opakovaně použitelný i pro následující ročníky. Během vývoje jsem také kladl na důraz zpracovávání v reálném čase – každý uživatel může vidět, kolik je na které přednášce přihlášených lidí a hlavně jestli se uživatel na přednášku úspěšně přihlásil nebo mu místo uteklo a musí se přihlásit jinam.

3. Hlavní funkce

3.1. Jednotné přihlašování do systému

Jednou z absolutních priorit projektu bylo všem uživatelům umožnit přihlašování do systému přes školní Google účty. Pomocí protokolu OAuth 2.0 se uživatelé mohou do systému bezpečně přihlašovat. Samozřejmostí je i kontrola domény e-mailové adresy, která slouží k zamezení přihlašování lidí mimo naše gymnázium.

Velkou tradicí Plejád je ovšem i to, že se na ně přijdou podívat i absolventi školy, kteří již nemají školní e-mail. Správci systému tedy mohou nastavit omezený počet hostů, kteří pouze zadají e-mail, který nesmí být školní, do odděleného formuláře. Jsou tak odděleni od vlastního přihlašování, ale organizátoři s jejich počtem mohou počítat.

3.2. Přihlašování k přednáškám

Hlavním cílem systému je umožnit studentům se jednoduše a pohodlně přihlásit k přednáškám. Vzhledem k tomu, že přihlašování funguje na principu „kdo dřív přijde, ten má větší šanci, že se dostane na přednášku“, musí být uživatelské rozhraní co nejjednodušší a co nejvíce intuitivní. Vzhledem k fungování přihlašování je kladen důraz na zobrazování kapacity přihlášených v reálném čase, aby uživatel věděl, zdali je možné se na přednášku ještě přihlásit.

3.3. Administrátorské rozhraní

Organizátorský tým musí zvolit jednoho správce systému, který se následně postará o celé nastavení systému. Nejedná se ale o náročný úkol a dá se zvládnout během pár desítek minut. Hlavní správce je schopen přidávat a upravovat přednášky pomocí jednoduchých formulářů. Celé administrátorské rozhraní vypadá velice podobně jako rozhraní pro přihlašující a je tedy schopen vše vidět s co největší přehledností.

V administrátorském rozhraní se také přidávají ostatní organizátoři, kteří nejsou schopni systém administrovat a ani jim není dovoleno se na přednášky přihlašovat – jde pouze o evidenci pro vedení školy kvůli správnému vyřešení absencí. Systém dále umožňuje export všech přihlášených studentů pro tisk – tyto formuláře musí být rozneseny po třídách, ve kterých se přednášky konají. Stejně jako u ostatních organizátorů jde čistě o evidenci a podepsané archy jsou určeny pro vedení školy.

4. Technologie

Celý systém byl vyvinutý v programovacím jazyce Elixir. Jedná se o dynamický funkcionální programovací jazyk běžící na architektuře BEAM (známé též jako Erlang VM). Jde o alternativní syntax a nástavbu pro programovací jazyk Erlang, který se již od 80. let využívá v telekomunikacích a je známý pro svou schopnost masivní paralelizace a odolnosti proti selhání (fault-tolerant system). Protože byl při vývoji systému Plejády kladen velký důraz na schopnost vyhodnocování v reálném čase a na velkou odolnost vůči velkému počtu uživatelů využívající systém v jednu chvíli, se Elixir se svou robustní architekturou se základy v telekomunikacích jevil jako příhodný nástroj.

Vedle samotného programovacího jazyka velkou část aplikace pokrývá webový framework Phoenix. Ten zajišťuje veškerou obousměrnou komunikaci mezi klientem a serverem a umožňuje vývojáři jednoduše implementovat uživatelské rozhraní, ve kterém se data mění v reálném čase.

5. Závěr

Celý maturitní projekt považuji za zdařilý. Implementace hlavních funkcí, které byly podmínkou, proběhla bezproblémově a celý systém je připravený na použití a k dispozici pro budoucí ročníky. Systém byl otestován na letošním ročníku Plejád, pro který byl během posledních dvou týdnů září 2022 bez jakýchkoli funkcí pro administrátory vyvinut jako prototyp tohoto maturitního projektu. V jeden moment systém využívalo 350 studentů a systém bezchybně zaregistroval všechny. Během práce na tomto projektu jsem přidal veškeré administrátorské rozhraní a zoptimalizoval samotné přihlašování pro studenty. Také jsem se zdokonalil ve funkcionálním programování a vývoji fault-tolerant systémů a dozvěděl se něco nového o distribuovaných systémech, což je téma, které mě zajímá a chtěl bych se mu v budoucnu věnovat.

6. Odkazy

- Demoverze (<https://plejady.marrt.in>)
- Dokumentace (<https://rosenbergm.github.io/plejady>)
- Zdrojový kód (<https://github.com/rosenbergm/plejady>)
- Uživatelská příručka (https://rosenbergm.github.io/plejady/user_manual.html)

7. Zdroje

7.1. Použité knihovny

- **elixir** (<https://hexdocs.pm/elixir>)
- **phoenix** (<https://hexdocs.pm/phoenix>)
- **phoenix_ecto** (https://hexdocs.pm/phoenix_ecto)
- **ecto_sql** (https://hexdocs.pm/ecto_sql)
- **postgrex** (<https://hexdocs.pm/postgrex>)
- **phoenix_html** (https://hexdocs.pm/phoenix_html)
- **phoenix_live_reload** (https://hexdocs.pm/phoenix_live_reload)
- **phoenix_live_view** (https://hexdocs.pm/phoenix_live_view)
- **floki** (<https://hexdocs.pm/floki>)
- **phoenix_live_dashboard** (https://hexdocs.pm/phoenix_live_dashboard)
- **esbuild** (<https://hexdocs.pm/esbuild>)
- **tailwind** (<https://hexdocs.pm/tailwind>)
- **swoosh** (<https://hexdocs.pm/swoosh>)
- **finch** (<https://hexdocs.pm/finch>)
- **telemetry_metrics** (https://hexdocs.pm/telemetry_metrics)
- **telemetry_policy** (https://hexdocs.pm/telemetry_policy)
- **gettext** (<https://hexdocs.pm/gettext>)
- **jason** (<https://hexdocs.pm/jason>)
- **plug_cowboy** (https://hexdocs.pm/plug_cowboy)
- **ueberauth** (<https://hexdocs.pm/ueberauth>)
- **ueberauth_google** (https://hexdocs.pm/ueberauth_google)
- **cachex** (<https://hexdocs.pm/cachex>)
- **ex_cldr** (https://hexdocs.pm/ex_cldr)
- **ex_cldr_plugs** (https://hexdocs.pm/ex_cldr_plugs)
- **ex_cldr_dates_times** (https://hexdocs.pm/ex_cldr_dates_times)
- **tz** (<https://hexdocs.pm/tz>)
- **credo** (<https://hexdocs.pm/credo>)
- **ex_doc** (https://hexdocs.pm/ex_doc)
- **timex** (<https://hexdocs.pm/timex>)

7.2. Další použitá dokumentace

- <https://elixir-lang.org/docs.html>