GYMNASIUM JANA KEPLERA

Parléřova 2/118, 169 00 Praha 6



Webové stránky pro Chytrou palici

Maturitní práce

Autor: Michal Vaňata

Třída: 4.A

Školní rok: 2020/2021

Předmět: Informatika

Vedoucí práce: Šimon Schierreich



GYMNASIUM JANA KEPLERA Kabinet informatiky

ZADÁNÍ MATURITNÍ PRÁCE

| Student: | Michal Vaňata | |
|--|---|------|
| Třída: | 4.A | |
| Školní rok: | 2020/2021 | |
| Platnost zadání: | 21. 10. 2021 | |
| Vedoucí práce: | Šimon Schierreich | |
| Název práce: | Webové stránky pro Chytrou palici | |
| autorizovanému u | ání: portál pro školní literární soutěž Chytrá palice. Portál bu uživateli umožňovat dynamické přidávání a editaci obsal Concový uživatel pak bude mocu obash filtrovat, vyhledáv | hu a |
| 2000. Dostupné z: principles.pdf. [2] FOWLER, Mart | a: ert C. Design Principles and Design Patterns. www.objectmes thttps://fi.ort.edu.uy/innovaportal/file/2032/1/des tin. Patterns of enterprise application architecture. Boston: A al, 2003. ISBN 978-0321127426. | ign_ |
| URL repozitáře: | | |
| https://github.c | com/Daybringer/chytra-palice | |
| | | |
| vedoucí práce | student | |

| Prohlášení |
|---|
| Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze prameny a liteaturu uvedené v seznamu bibliografických záznamů. Nemám žádné námitky proti zpřítupňování této práce v souladu se zákonem č. 121/2000 Sb. o právu autorském, o právech ouvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů. |
| Praze dne 7. dubna 2021 Michal Vaňata |

| Poděkování | |
|--|--------------|
| | |
| Chtěl bych poděkovat vedoucímu maturitní práce, Šimonu Schierreichovi za poskytnuté rady a konzultace. | é materiály, |
| | |
| | |

Abstrakt

V práci popisuji, jak jsem postupoval při vypracování maturitního projektu. Popisuji problematiku vývoje webové aplikace. V uživatelském manuálu názorně zobrazuji způsoby využívání základních funkcí stránky.

Klíčová slova

webová aplikace, CMS, správa soutěže

Abstract

In this document I try to describe how I dealt with the development of this project. I explore and explain the problems and solutions of software development. In the user's manual I describe and explain features of Chytrá palice web app.

Keywords

web application, CMS, contest management

Obsah

| 1 | Teor | etická č | ást | 3 | | | |
|----|------|--------------|----------------------------------|----|--|--|--|
| 2 | Imp | Implementace | | | | | |
| | 2.1 | Před v | ývojem | 5 | | | |
| | | 2.1.1 | Sběr požadavků a tvorba slovníku | 5 | | | |
| | | 2.1.2 | Tvorba diagramů | 5 | | | |
| | 2.2 | Vývoj . | | 7 | | | |
| | | 2.2.1 | Frontend | 7 | | | |
| | | 2.2.2 | Server | 8 | | | |
| 3 | Tech | nická d | okumentace | 9 | | | |
| • | 3.1 | Instala | ce a spuštění | 9 | | | |
| | 3.2 | | elský manuál | 13 | | | |
| | | 3.2.1 | Přidání admina | 13 | | | |
| | | 3.2.2 | Hlavní panel | 13 | | | |
| | | 3.2.3 | Články | 15 | | | |
| | | 3.2.4 | Soutěž | 17 | | | |
| | | 3.2.5 | Práce | 18 | | | |
| Zá | věr | | | 21 | | | |
| Se | znam | použité | literatury | 23 | | | |
| Se | znam | obrázk | ů | 25 | | | |
| Se | 7nam | tahulek | | 26 | | | |

1. Teoretická část

Cílem projektu je vytvoření nové webové platformy pro školní literární soutěž Gymnázia Jana Keplera, Chytrou palici. Původní webová stránka sloužila pouze k nahrávání, filtraci a zobrazování prací zařazených do soutěže. Její vzhled byl zastaralý a některé její části byly nefunkční, a proto byl vypsán projekt na tvorbu nové. Vzhledem k tomu, že se stránka navrhovala od úplných základů, se naskytla možnost obohatit stávající a přidat další funkce.

Výsledná webová stránka by nakonec krom původních funkcí měla umožňovat přidávání a editaci článků blogového typu, vytváření soutěže, vyhlášení závěrečného pořadí, nahrávání obrázků a konverzi a následné stažení nahraných soutěžních prací. Jedním z dalších požadavků bylo umožnění komentování prací studenty.

2. Implementace

Vznik každé větší webové aplikace sestává z dvou částí: návrhu a programování. Návrhová část se v mém případě skládala ze sběru požadavků zadavetele, tvorby společného jazyka (*ubiquitous language*) a tvorby diagramů a datových modelů. Při tvorbě datových modelů jsem již uvažoval nad přibližnou podobou a rozložením API přístupových bodů. Vzhledem k tomu, že je to poprvé, co tvořím webovou stránku takového rozsahu, tak jsem se musel dovzdělat v oblasti návrhu REST API. Dále jsem si musel rozmyslet strukturu stránek a vytvořit jejich přibližnou mapu.

Po návrhové části přichází implementace webové stránky, která se skládá z tvorby frontendu a backendu.

Při dobře provedené návrhové části bývá mnohdy samotné programování procházkou růžovým sadem, v případě podcenění přípravy naopak procházkou devíti kruhy Dantova Pekla.

2.1 Před vývojem

2.1.1 Sběr požadavků a tvorba slovníku

První krokem při tvorbě Chytré palice byly konzultace se zadavetelem projektu, při kterých jsem si musel ujasnit, co od stránky očekává, jaké má připomínky k podobě a funkcionalitě starých stránek a jak by si představoval přibližnou podobu. Po konzultacích by měla již být vytvořená tabulka požadavků (v 2.1), s kterou bude nadále pracovat (např. UCD).

Dále by jsme měli mít vytvořený tzv. *ubiquitous language*, což si můžete představit jako vytvoření slovníčku termínu, které budeme společně se zákazníkem používat. Jde o to, aby mezi zákazníkem a vývojářem nedocházelo k miskomunikacím. Inspirací při tvorbě designu a rozložení stránky mi měli být již zažité webové platformy A2larm a A2.

2.1.2 Tvorba diagramů

Dalším krokem je vytvoření diagramů.

Ty tvoříme především pro ujasnění struktury programu a dat v něm.

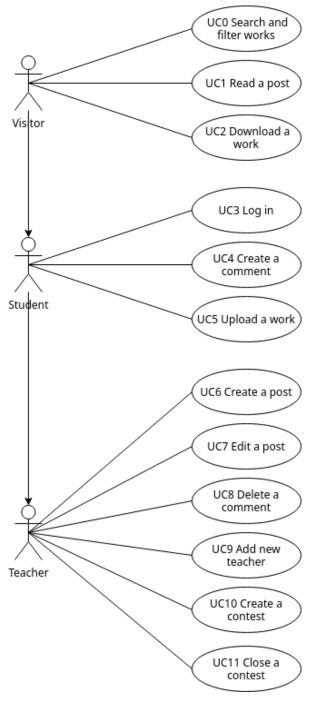
Pro tvorbu diagramů jsem použil webový nástroj DrawIO dostupný na https://app.diagrams.net

Use case diagram

Use case diagram neboli diagram užití je typem UML diagramu, který se používá jako náhled na programovaný systém. Zobrazují se v něm vztahy mezi tzv. aktory čili uživateli systému a službami. Může mít jak textovou podobu, která bývá podrobnější a každý případ užití je podrobně popsán, tak i podobu diagramovou.

Je u něj důležitá jednoduchost, neboť se primárně používá pro komunikace mezi vývojářem a zákazníkem. To znamená, že by neměl obsahovat technikálie a měl by používat tzv. *ubiquitous language*.

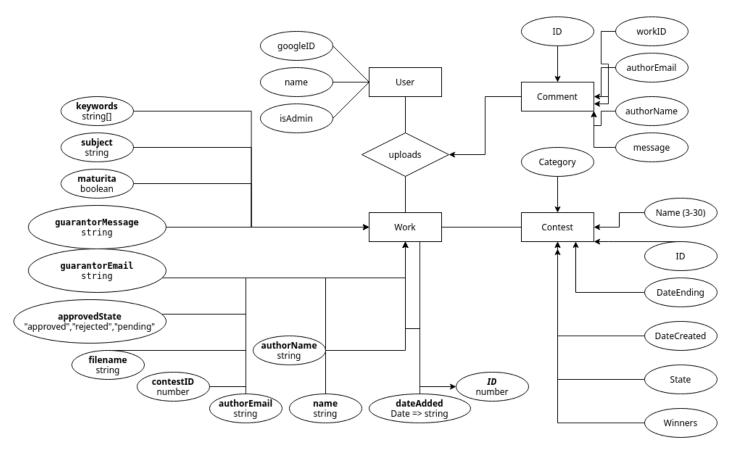
Při jeho vytváření využíváme tabulku požadavků, abychom si byli jistí, že jsme splnili všechny funkční požadavky.



Obrázek 2.1: UCD

Entity-relationship diagram

Entity-relationship diagram je vizuální reprezentací objektů (entit) a vztahů mezi nimi. ER diagramy se nejčastěji používají při analýze informačních systémů a u návrhů systémů při jejich budování. Při správném vytvoření je dokonce možné použít jeho variantu pro vygenerování databáze.



Obrázek 2.2: ERD

2.2 Vývoj

Jakmile jsme hotovi s přípravou, tak se můžeme pustit do vývoje.

V mém případě jsem si jako první připravil frontendovou část webové aplikace, k níž jsem později připojil server. Nakonec bych zvolil asi cestu opačnou, ale o tom více v závěru. Ideálně by celý proces vývoje mělo doprovázet psaní testů, aby jsme si byli jisti, že jsme u psaní splnili určitě nároky na kód.

Při volbě technologií jsem především zohledňoval svojí předchozí zkušenost s ní a až poté jsem uvažoval nad vhodností knihovny či nástroje.

2.2.1 Frontend

Vzhledem k rozsáhlosti projektu byla nasnadě volba některého javascriptího frameworku, abych si ulehčil práci s opakujícími se prvky a s reaktivitou dat. Zde se v podstatě nabízejí 4 možnosti, a to React, VueJS, Svelte a Angular. Z těchto frameworků jsem měl dosavadní zkušenost jen s Vue a jelikož jsem chtěl ihned začít tvořit prototyp a mezi ostatními frameworky není takový rozdíl, tak jsem zvolil jej. U Vue existují v tuto dobu dvě paralelní verze: 2 a 3. Verze 3 nabízí řadu zlepšení v reaktivitě dat a hlavně bezproblémové používání Typescriptu. Velkou nevýhodou je ale nezralost ekosystému knihoven, což nakonec rozhodlo o mém použití VueJS verze 2.

Při vytvoření prototypu bylo nutné simulovat volání na server (REST API), pro tento účel jsem využil mockAPI.

Abych nemusel věnovat přespřílišnou pozornost stylům a tvorbě UI, tak jsem využil předpřipravené komponenty z Buefy.

K voláním na server jsem použil nejpoužívanější knihovnu, axios, který jsem za pomocí repository design patternu a Vuex (Vue store) implementoval tak, aby byla možná jednoduchá úprava konfigurací a přístupových bodů.

2.2.2 Server

Pro jazyk použitý na serveru jsem měl dvě podmínky, a to aby byl typovaný a umožňoval snazší objektové programování. Vzhledem k předchozí znalosti Javascriptu jsem zvolil Typescript. Dále jsem musel najít framework a databázové ORM, které by jej nativně podporovali. NestJS a TypeORM s Postgresql byly nakonec mou volbou.

Pro konverzi nahraných soutěžních prací z otevřených formátů (.doc, .docx, .odt) do pdf jsem využil funkci konverze souborů programu LibreOffice.

| Název požadavku | Popis požadavku | Fn |
|--------------------|--|-----|
| Filtrování | práce a soutěže by měli být filtrovatelné na základě vlastností | ano |
| Uspořádávání | práce a soutěže by měli být uspořadatelné na základě vlastností | ano |
| Zobrazení práce | návštěvník si může práci zobrazit skrze prohlížeč | ano |
| Stažení práce | návštěvník si může práci stáhnout ve formát PDF či EPUB | ano |
| přihlášení | studenti a učitelé by měli být schopní přihlásit se přes školní mail | ano |
| Multimédia | umožnit přidání multimédií (obrázků) ke článku | ano |
| Práce | přidání soutěžní práce | ano |
| Soutěž | vytvoření a správa soutěže | ano |
| Adminy | zpřístupnit některé funkci jen adminům | ano |
| Migrace prací | přesunout staré práci na nový web | ne |
| Docker | před nahráním práce na školní web vytvořit docker image | ne |
| Reference na práci | umožnit vygenerování reference na práci při psaní článku | ano |
| Nahrání souboru | umožnit studentům nahrát práci ve formátech PDF, DOC(X), ODT | ano |
| Podcast | možnost dynamicky přidat odkaz na podcast | ano |
| Nahrání souboru | umožnit studentům nahrát práci ve formátech PDF, DOC(X), ODT | ano |
| Komentování | komentování soutěžní prací studenty a učiteli | ano |
| Nejčtenější | zobrazení nejčtenější práce | ano |

Tabulka 2.1: Tabulka požadavků

3. Technická dokumentace

3.1 Instalace a spuštění

LibreOffice

Pokud nemáte nainstalovaný LibreOffice, tak si jej nainstalujte https://www.libreoffice.org/get-help/install-howto

NodeJS

https://nodejs.org/en/download/

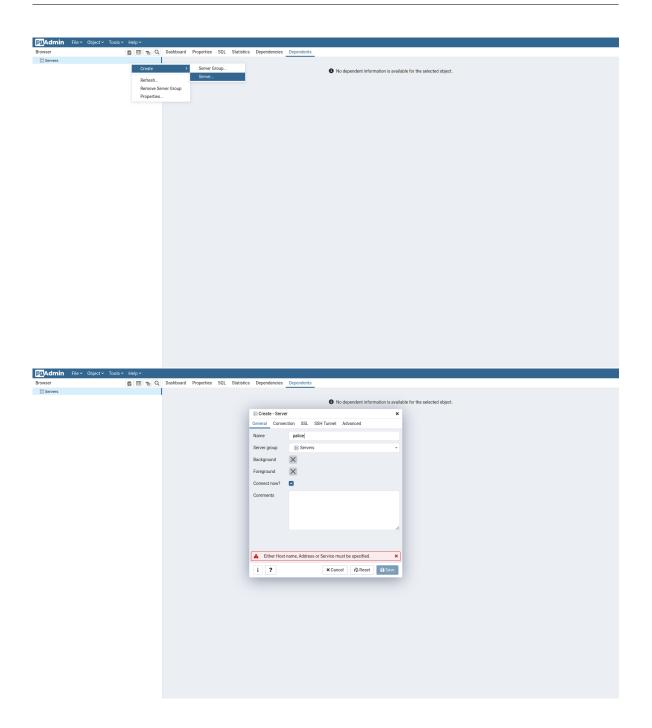
Yarn

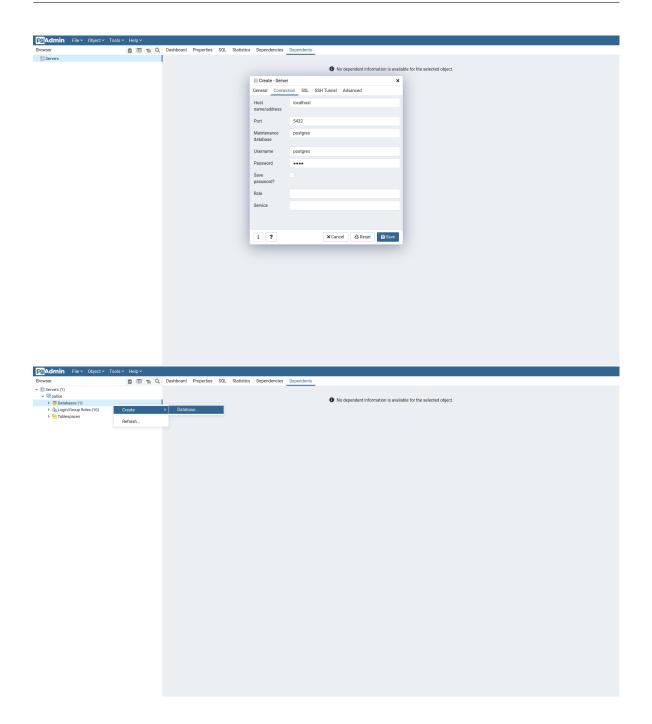
npm install --global yarn

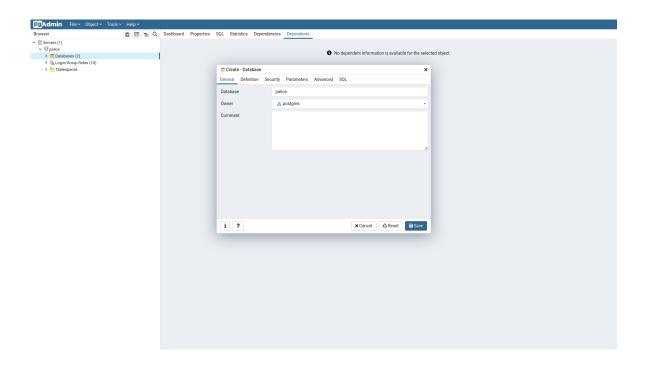
PostgreSQL a pgAdmin

Stáhněte a nainstalujte si PostgreSQL
https://www.postgresql.org/download/
Stáhněte a nainstalujte si pgAdmin (Není vyžadovaný pro chod aplikace, ale budu ukazovat postup
vytvoření databáze za jeho pomoci.)
https://www.pgadmin.org/download/

- 1. Spust'e pgAdmin
- 2. Vytvořte server se jménem **palice** a jako Hostname nastavte **localhost** s defaultním portem **5432** Jako Username nastavte **postgres** a jako heslo **root**
- 3. Vytvořte databázi s uživatelem **postgres** a pojmenujte ji **palice** Obrázkový návod:







Palice

Jestliže máte nainstalované a nakonfigurované všechny programy, tak můžete stáhnout a spustit samotnou webovou aplikaci.

```
git clone https://github.com/Daybringer/chytra-palice --recursive
cd chytra-palice
sh build.sh
```

Po úspěšné instalaci balíčků vytvořte ve složce **server** soubor *.env* s následujícími proměnnými:

```
GOOGLE_CLIENT_ID=922077049204-lh8mmn4bak6bj81v9r2n3ir522c2gqu3.apps.googleusercontent.com
GOOGLE_SECRET=tajnyUdajKteryVamZaslu
JWT_SECRET=VamiZvolenyNahodnyKlic
```

Poté z kmenové složky spusť te skript pojmenovaný start.sh, který se nachází v té samé kmenové složce.

sh start.sh

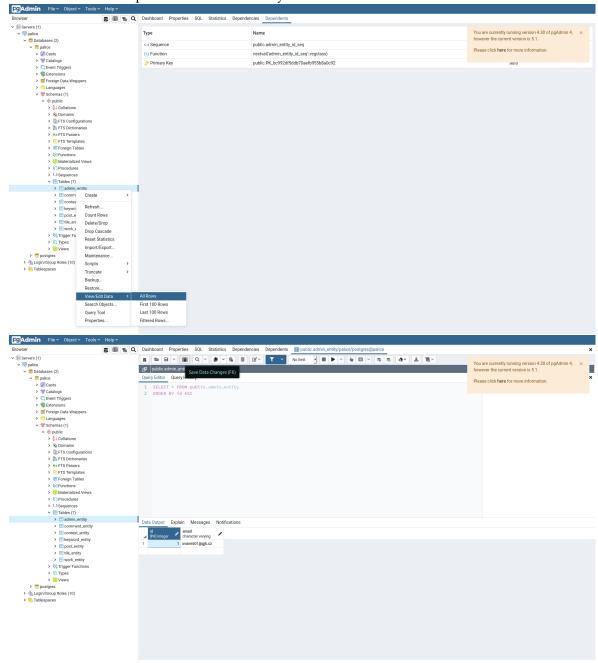
Stránka je dostupná na http://localhost:8080/

3.2 Uživatelský manuál

3.2.1 Přidání admina

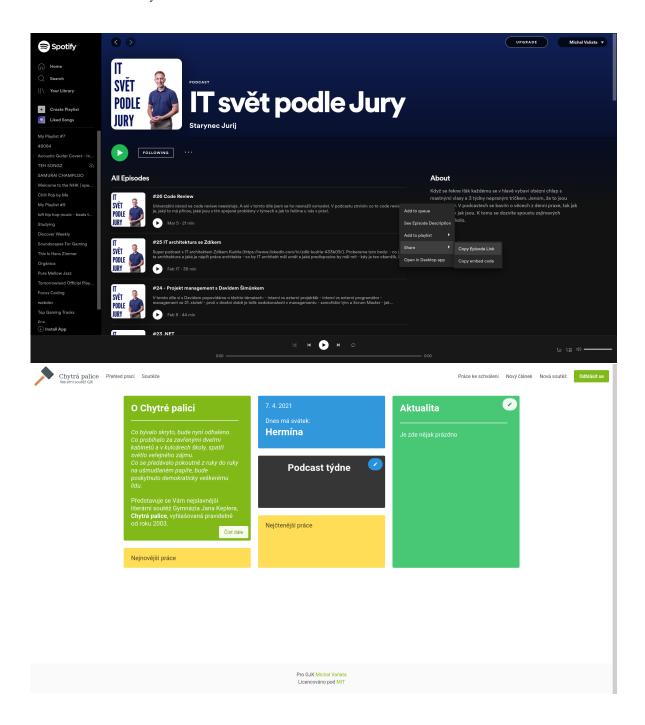
Otevřete si databázi palice v aplikaci pgAdmin.

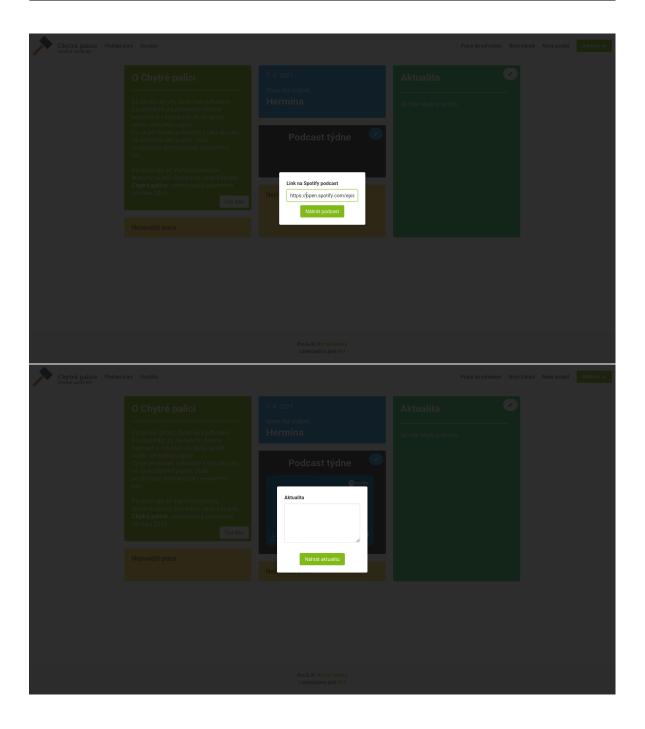
Dále si otevřete tabulku **admins** a vložte záznam s emailem uživatele, který se má stát adminem. Nakonec manuálně vpište unikátní numerický index a uložte.



3.2.2 Hlavní panel

Pokavaď je uživatel admin, tak může editovat obsah panelu s aktualitou a podcastem Spotify. Pro přidání podcastu musí admin zkopírovat link na epizodu a vložit jej do panelu. U aktuality stačí zadat text, který se má zobrazovat.



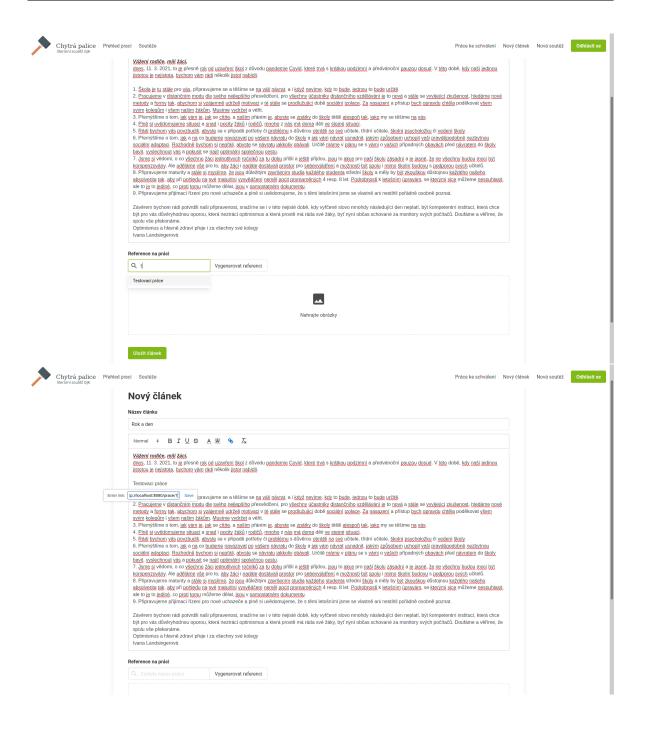


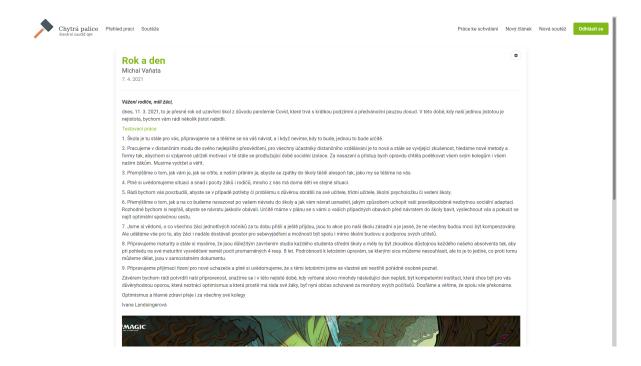
3.2.3 Články

Pokavaď je uživatel admin, tak může přidat nový článek, k čemuž se proklikne přes tlačítko **Nová soutěž** v navigační liště.

Vyplní název, obsah a eventuálně nahraje obrázky.

Jednou z funkcí je generování reference na konkrétní práci. Stačí práci vyhledat dle jejího názvu, kliknout na ní a kliknout na tlačítko **Vygenerovat referenci**. Poté stačí označit text v obsahu, označit jej jako hyperlink a vložit referenci.





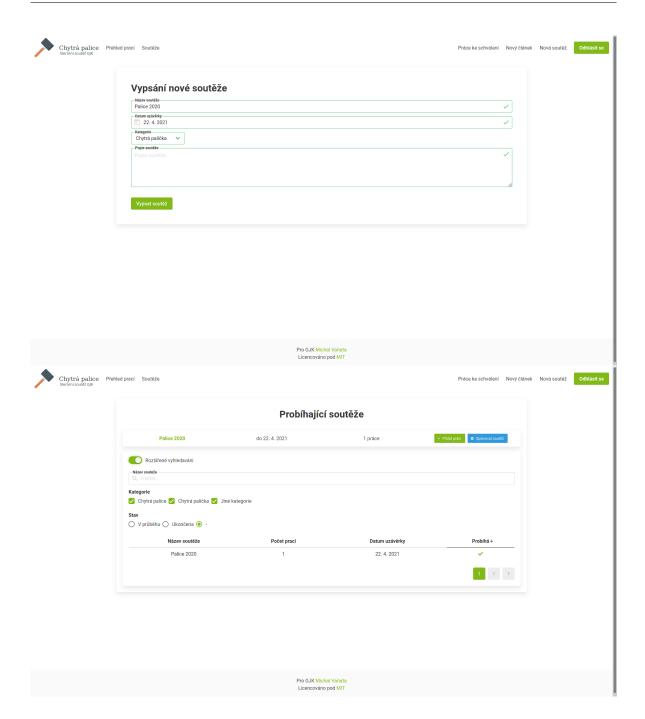
3.2.4 Soutěž

Administátor se může přes tlačítko **Nová soutěž** dostat k formuláři k jejímu založení. V něm vyplní její název, datum uzávěrky, kategorii a eventuálně její popis.

Po vypsání soutěže až do vypršení data úzávěrky nebo vyhlášení pořadí do ní studenti mohou přidávat soutěžní práce.

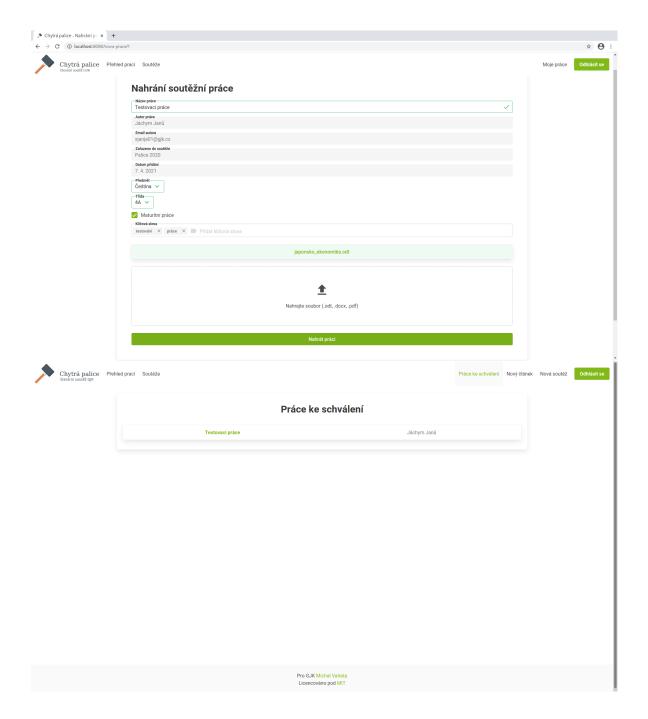
Jak soutěže tak práce je možné filtrovat dle různých atributů.

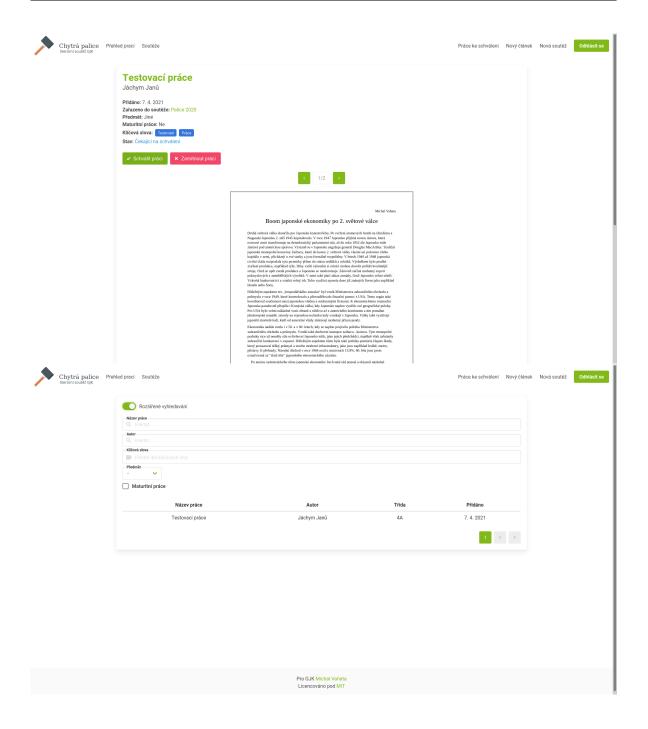
Práci admin uzavře přes profil práce, na který se dostane proklikem ze všech probíhající prací nebo z odkazu ze soutěžní práce. Uzavře jí kliknutím na tlačítko **Uzavření práce** a vybráním oceněných prací. Soutěž ale musí mít minimálně 3 nominované práce (schválené), jinak není možné obsadit všechny příčky.



3.2.5 Práce

Jak studenti, tak admini mohou přidávat soutěžní práci. Tam se dostanou přes profil soutěže a přes tlačítko přidat práci. Vyplní název práce, předmět, třídu, vloží klíčová slova a nahrají soubor. Práce je poté ve stavu *pending*, a aby se zobrazovala ostatním, tak musí být adminem schválena.





Závěr

Výsledkem mé práce je funkční webová aplikace pro účely školní literární soutěže. Na stránku je možné dynamicky přidávat obsah. Této editace se můžou účastnit jak studenti, tak učitelé na Gymnáziu Jana Keplera.

Se svou prací na projektu jsem byl vesměs spokojený. Mrzí mě, že jsem ke kódu nevytvořil řádné testy. Rozhodl jsem se tak kvůli velké náročnosti psaní testů v prostředí, kde dochází k dependency injection. Testovaní a dokumentace jsou určitě oblastmi, kterým bych rád u svého dalšího projektu věnoval více času. Dále nejsem spokojený s tím, jak jsem rozplánoval tvorbu projektu. Mnoho času jsem zbytečně strávil při designu některých diagramů, které jsem následně nevyužil, a tento čas jsem mohl věnovat jiným částem přípravy. Také bych začal s tvorbou serveru a ne frontendu.

Naopak jsem spokojený s tím, jak nakonec stránka vypadá a funguje, a jsem rád, že jsem splnil veškeré důležité požadavky zadavatele.

Seznam použité literatury

- [Fowo3] Martin Fowler. *Patterns of enterprise application architecture*. Boston: Addison-Wesley Professional, 2003.
- [Maroo] Robert C. Martin. *Design Principles and Design Patterns*. www.objectmentor.com, 2000.

Seznam obrázků

| 2.1 | UCD | | | | | | | | | | | • | | | | | | | | | | (|
|-----|-----|------|--|--|--|--|--|--|--|--|--|---|--|--|--|--|--|--|--|--|--|---|
| 2.2 | ERD | | | | | | | | | | | | | | | | | | | | | - |

Seznam tabulek

| 2.1 | Tabulka požadavků | | . 8 |
|-----|-------------------|--|-----|
|-----|-------------------|--|-----|