

GYMNASIUM JANA KEPLERA

Parléřova 2/118, 169 00 Praha 6



Webová stránka pro hodnocení obědů ze školní jídelny

Maturitní práce

Autor: Adam Dočkálek

Třída: 4.C

Školní rok: 2021/2022

Předmět: Informatika

Vedoucí práce: Šimon Schierreich

Praha, 2022



GYMNASIUM JANA KEPLERA *Kabinet informatiky*

ZADÁNÍ MATURITNÍ PRÁCE

Student: Adam Dočkálek
Třída: 4.C
Školní rok: 2021/2022
Platnost zadání: 30. 9. 2022
Vedoucí práce: Šimon Schierreich
Název práce: Webová aplikace pro hodnocení jídel ze školní jídelny

Pokyny pro vypracování:

Cílem práce je vytvořit webovou stránku, která slouží k hodnocení jídel ze školní jídelny. Na stránku by se měla automaticky posílat konkrétní jídla z jídelníčku, který se nachází na stránkách školy. Stránka by také měla umět třídit určitá jídla do různých kategorií (např. zda-li jsou vegetariánská) a to buď automaticky, či manuálně. Každé jídlo by mělo mít stránku, kde je možné psát komentáře, fotografie a hodnocení.

Doporučená literatura:

- [1] MARTIN, Robert C. *Design Principles and Design Patterns*. www.objectmentor.com, 2000. Dostupné z: https://fi.ort.edu.uy/innovaportal/file/2032/1/design_principles.pdf.
- [2] FOWLER, Martin. *Patterns of Enterprise Application Architecture*. Boston, Massachusetts, USA: Addison-Wesley Professional, 2003. The Addison-Wesley Signature Series. ISBN 978-0-321-12742-6.
- [3] EVANS, Eric. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston, Massachusetts, USA: Addison-Wesley Professional, 2003. ISBN 978-0-32-112521-7.
- [4] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

URL repozitáře:

<https://github.com/Docksein/Maturitni-prace>

student

vedoucí práce

V Praze dne 22. 10. 2021

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Nemám žádné námitky proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praze dne 25. března 2022

Adam Dočkálek

Poděkování

Chtěl bych poděkovat vedoucímu práce Šimonu Schierreichovi za rady, které pomohly, aby práce byla funkční.

Abstrakt

Práce má za účel vytvořit webovou aplikaci, na které je možné hodnotit obědy ze školní jídelny Gymnázia Jana Keplera. K vytvoření aplikace byl použit programovací jazyk Python a webový framework Django, který umožňuje relativně snadnou tvorbu webových aplikací s velkou podporou. Na automatické posílání dat z webové stránky jídelny byla použita knihovna request, která umožňuje brát data z webových stránek a knihovna BeautifulSoup4, která dokáže filtrovat HTML soubory. Výsledkem práce je webová aplikace, která umožňuje přihlášeným uživatelům psát hodnocení a komentáře ke školním obědům, které jsou adminem automaticky posílány do databáze. Uživatel si také může prohlédnout vlastní hodnocení, která již napsal.

Klíčová slova

webová aplikace, hodnocení, obědy

Abstract

This work is aimed to create a web application, on which a user can review lunch foods from the school canteen of Grammar School of Johannes Kepler. To create this application the programming language that was used is Python and the web framework Django, which allows relatively easy development of web application with a large amount of support. The automatic pulling of data from the website of the school canteen uses the library request that can get data from a website and the library BeautifulSoup4, that can filter through HTML files. The result of this work is a web application that allows rating and commenting of lunch foods for logged-in users. The foods are automatically saved to database with the help of an admin user. The user can also view all of his past reviews.

Keywords

web application, reviews, lunches

Obsah

1	Teoretická část	3
1.1	Hodnocení	3
1.2	Přihlašování uživatelů	3
1.3	Automatické přidávání jídel	3
2	Implementace	5
2.1	Databáze	5
2.2	Registrace a přihlašování	6
2.3	Hodnocení a komentáře	6
2.4	Automatické posílání jídel	7
3	Technická dokumentace	9
	Závěr	11
	Seznam obrázků	13
	Seznam tabulek	14

1. Teoretická část

Cílem práce je vytvoření webové aplikace, na které je možné hodnotit jídla ze školní jídelny. Měla by mít několik funkcí ke kterým samozřejmě patří hodnocení jídel, psaní komentářů a přihlášení uživatelů, kteří mohou hodnocení psát. Samotná jídla by se měla automaticky přidávat do databáze.

1.1 Hodnocení

Stránka by měla mít seznam jídel a každé jídlo by mělo mít možnost zapsat komentář a hodnocení. Jídla mají název a hodnocení, na stránce by se měl zobrazovat průměr hodnocení od 1 do 5 a z této škály si vybírá uživatel. Dále pro uložení komentáře musí uživatel napsat vlastní přezdívku a volitelně může napsat komentář k jeho hodnocení. Pokud byl komentář uložen, uživatel bude přesměrován a komentář se zobrazí pod popisem daného jídla, ke kterému komentář patří.

1.2 Přihlašování uživatelů

Anonymní uživatel by měl možnost zobrazit si všechna jídla i jejich detaily(komentáře, hodnocení, apod.), ale neměl by mít možnost zapisovat komentáře, aby nebylo komentáře možné spamovat. Samotná registrace a přihlášení by mělo být relativně základní, stačí pouze jméno a heslo, které si uživatel zvolí a po registraci by měl mít již přístup k vytvoření komentáře.

1.3 Automatické přidávání jídel

Vzhledem k tomu, že jsou jídla přidávána každý týden na stránce školy, tak by mělo být možné data tahat a přidávat je do databáze pomocí programu automaticky.

2. Implementace

Pro vytvoření webové aplikace jsem použil knihovnu Django pro jazyk Python. Učinil jsem tak hlavně z toho důvodu, že je to jeden z nejpoužívanějších jazyků a má rozsáhlou podporu a komunitu, takže je možné najít odpověď na otázky ohledně případných errorů, které mohou nastat. Vzhledem k tomu, že jsem dělal webovou aplikaci poprvé v životě, tak bylo jisté, že se dopustím spousty chyb a v tomto případě je relativně jednoduché najít odpověď. Django má také spoustu funkcí a zabudovaných aplikací, které je možné využít (např. aplikace na registraci a přihlašování uživatelů, kterou v práci využívám). Pro stylizování front-endu jsem použil framework Bootstrap, který mi byl doporučen vedoucím práce.

2.1 Databáze

Pro webovou aplikaci je potřeba databázi s dvěma tabulkami: tabulka Jídel, která má název, datum, kdy byla položka přidána a komentář k přidání. Druhá je tabulka komentářů, která skladuje data jmen uživatelů, jednotlivá hodnocení (v tomto případě od jedné do pěti), datum přidání komentáře a klíč k jídlu, ke kterému patří daný komentář.

Nejdříve bylo nutné vytvořit samotnou aplikaci na komentáře, která obsahuje předem vytvořené soubory, do kterých se píšou třídy a funkce, které potom komunikují s databází a front-end html soubory.

Architektura Djanga je založená na MVC, takže přístup k databázi zajišťují modely, které jsou v django aplikaci pod souborem models.py. V praxi pak implementace vypadá následovně:

```
class Food (models.Model):
    title = models.CharField(max_length=100)
    upload_date = models.DateTimeField(default=timezone.now)
    description = models.CharField(max_length=100, null=True )

    def average_rating(self):
        all_ratings = map(lambda x: x.ratings, self.review_set.all())
        return str(numpy.mean(list(all_ratings)))[5]
    def get_absolute_url(self):
        return reverse("food_reviews", kwargs={"food_id": self.id})
    def __str__(self):
        return self.title

class Review (models.Model):
    rating_choices = (
        (1, 1),
        (2, 2),
        (3, 3),
        (4, 4),
        (5, 5) )
```

```
published_date = models.DateTimeField(default=timezone.now)
food_key = models.ForeignKey(Food, on_delete=models.CASCADE)
author_name = models.ForeignKey(User, on_delete=models.CASCADE)
comment = models.CharField(max_length=1000)
ratings = models.IntegerField(choices=rating_choices)
```

Dále v kódu můžeme vidět i některé funkce, funkce `average_rating()` slouží k vypsaní průměru hodnocení, která je použita v html souborech, který používá django formy. Stejně tak i funkce vypisující název jídla je poté využita v zobrazení na webové stránce.

2.2 Registrace a přihlašování

Pro registrování je použita věstavná aplikace 'django.contrib.auth', která je určená na zvládnutí jak registrací, tak i přihlášení. Má zabudované modely pro registraci i přihlášení a automaticky vytvoří tabulky s databází, kam se zapisují jména a hesla. Aplikace nabízí i více možností pro jiná pole jako například pro email, ale myslím si, že nebylo nutné je pro tento projekt zahrnovat a stačilo pouze jméno a heslo.

2.3 Hodnocení a komentáře

Aby bylo formulář možné nechat renderovat na prohlížeči, je nejdříve nutné si vytvořit formu s poli v souboru `forms.py`, které chceme, aby vyplnil uživatel, kterými jsou hodnocení, které je povinné a komentář, který je nepovinný. Tato forma je poté vyrenderována a uživatel vidí formulář se dvěma poli, které by měl vyplnit. Aby nemohl nepřihlášený návštěvník stránky také psát do databáze, je tento formulář vyrenderovaný jen pro přihlášené uživatele díky zabudované funkci `user.is_authenticated`, pro nepřihlášené uživatele je pouze vidět text „Pro hodnocení se musíte přihlásit“.

Po úspěšném přidání hodnocení je uživateli zobrazena stránka s textem „Váš komentář byl úspěšně zapsán“. Veškerá hodnocení pro dané jídlo jsou zobrazeny pod jídlem a formulářem pro poslání hodnocení. Je zobrazeno jméno autora, jeho hodnocení, datum uložení komentáře a případný komentář autora.

Při implementaci jsem narazil na problém, že při snaze uložit komentář stránka uživatele přesměrovala na seznam jídel a přesto že request na POST přišel zpět na back-end, hodnocení se v databázi neuložilo. Důvodem bylo, že request směřoval do jiné view funkce, která renderovala stránku se seznamem jídel. Ovšem tato funkce již nezvládala POST requesty z front-endu a tím pádem se již neukládaly do databáze. Důvodem byla akce form v html, která automaticky přesunula POST právě funkci renderovací seznam jídel. Proto bylo nutné POST přesměrovat zpět do funkce, která zvládala zapsat komentář do databáze.

Uživatel má také možnost podívat se na svá hodnocení v sekci „Moje hodnocení“, ovšem nepodařilo se mi, aby měl možnost vlastní hodnocení měnit či smazat, to bohužel umí pouze admin.

2.4 Automatické posílání jídel

Pro automatické posílání jídel jsem využil knihoven `beautifulsoup4` a `request`. Pomocí funkce `request.get()` jsem vzal data ze stránky <https://gjk.cz/o-skole/jidelna/jidelnicek/>, kde jsem našel ve zdroji odkaz na tabulku se všemi týdenními jídly. Sice je možné podat request i na samotné stránce GJK, tak je jednodušší vzít pouze tabulku. Pomocí funkce `soup.find_all()` jsem našel všechny instance, které obsahovaly jídla. Ovšem musel jsem ještě odfiltrovat alergeny, které měly stejné třídy a prázdné kolonky. Bohužel z nějakého důvodu jsou na dvou školních programech napsány alergeny u jídel, takže tyto jídla se do databáze nedostanou.

Prvním plánem bylo vytvořit crone job pro django, který bude jednou týdně spouštět tuto funkci a následně zapisovat jídla do databáze, ovšem nepodařilo se mi crone job zprovoznit a django tento crone job nespouštělo. Proto jsem vytvořil funkci pro účty správců, kteří napíší pouze komentář (například jaký týden vzali jídla) a po kliknutí na „Přidat jídla“ se již jídla pomocí funkce na tahání jídel ze školní stránky uloží do databáze.

3. Technická dokumentace

Pro stažení repozitáře do vlastního zařízení lze stáhnout repozitář přímo z Github.com nebo použít tento příkaz:

```
git clone https://github.com/Docksein/Maturitni-prace
```

Pro spuštění souboru je potřeba knihovny, které stáhnete následovně:

```
pip install Django beautifulsoup4 request
```

Inicializování databáze:

```
python manage.py migrate
```

Spuštění:

```
python manage.py runserver
```


Závěr

Dovolil bych si tvrdit, že je zadání z větší části splněné. Myslím si že část jako tagy nebyla pro stránku s jídly potřebná, ovšem uvědomil jsem si to až po schválení zadání, ovšem hlavní body zadání splněny byly. Osobně jsem se naučil práci s Django a tvořením webových aplikací, protože jsem na žádném takovém projektu nikdy předtím nepracoval. Naučil jsem se jak fungují databáze v praxi a propojení databází, jak ukládat data z front-endu na back-end a jak potom data z databází zobrazit zpátky na webovém prohlížeči a aby byla data z jedné databáze propojené s daty v druhé na určité stránce. Osobně jsem se snažil, aby práce úspěšně naplnila zadání a abych se naučil nové věci, kterých bych do budoucna mohl využít.

Seznam obrázků

Seznam tabulek