

Gymnázium, Praha 6, Arabská 14

Arabská 14, Praha 6, 160 00



Ročníková Práce

Předmět : Programování

Téma : Žeryk - Sociální Síť pro sdílení receptů

Autor : Mariia Gavrylenko, Matěj Kratochvíl, Timon Eiselt

Třída : 3.E

Školní rok : 2023/24

Vyučující : Mgr. Jan Lána

Třídní učitel : Mgr. Blanka Hniličková

Prohlašujeme, že jsme jedinými autory tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů udělujeme bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne

Anotace

Výsledkem tohoto ročníkového projektu je sociální síť, jejímž cílem je umožnit lidem sdílet svoje oblíbené recepty pokrmů s ostatními nebo právě naopak hledat a objevovat nové dosud jimi nepoznané recepty. Uživatel je nejprve povinen se registrovat se svými osobními údaji a až poté bude oprávněn se přihlásit za pomoc svých přihlašovacích údajů (uživatelské jméno + heslo). Sociální síť bude umožňovat vše co by mohlo být potřeba, ať už od vyhledávání receptů, filtrace receptů s vybranými ingredienty, interakce s ostatními publikovanými recepty, tak po vlastní vkládání receptů.

Abstract

The result of this year's project is a social network that allows people to share their favourite recipes with others or, on the contrary, to search for and discover new recipes that they have never tried before. The user must first register with his details and only then can he log in with his credentials (username + password). The social network will allow everything from searching for recipes, filtering recipes with selected ingredients, interacting with other published recipes to posting recipes.

Obsah

1. Zadání projektu	1
2. Úvod	2
3. Technologie	3
3.1. Django	3
3.2. Uživatelské rozhraní	3
3.3. Databáze	4
3.4. Dodatečné knihovny	5
3.4.1. Pillow	5
3.4.2. Requests	5
3.4.3. Django Crispy Forms a Crispy-Bootstrap4	6
3.4.4. Django-ckeditor a Django-richtextfield	7
3.4.5. Django-modeltranslation	7
4. Scény aplikace	8
4.1. Registrace	8
4.2. Přihlášení	9
4.3. Domovská stránka	10
4.4. O nás	11
4.5. Vytváření receptů	12
4.6. Detaily receptů	14
4.8. Vyhledávání receptů	16
4.9. Profil	17
4.10. Oblíbené recepty	18
4.11. Nastavení	19
4.12. Změna hesla	20
5. Instalace, setup a spuštění	21
6. Závěr	22
7. Zdroje	23
8. Zdroje využité při programování	24
9. Seznam Obrázků	25

1. Zadání projektu

Zadáním projektu bylo vytvořit sociální síť, ve které si uživatel bude moci prohlížet a publikovat recepty (fotky, ingredience, návod). Taky bude moci dát like a nechat nějaký komentář. Před tím se ale musí zaregistrovat nebo se přihlásit. V profilu si uživatel najde informaci o svém účtu a budou tam zobrazeny jeho vlastní recepty. Recepty bude možné také vyhledávat a filtrovat podle různých kritérií (ingredience).

Pro vytvoření naší aplikace jsme použili Django, Bootstrap, HTML a CSS.

2. Úvod

Při brainstormingu na téma našeho ročníkového projektu našimi hlavami prošlo mnoho nápadů a lákavých možností, ale všichni jsme se s chutí přiklonili pro nápad vytvoření sociální sítě s recepty. Dali jsme dohromady 2 nejpopulárnější věci nejen u studentů, ale i ostatních sociální sítě a jídlo. Každý snad ví, že Česko je velmi spořivý národ, kdo by také nerad ušetřil sem tam nějaký ten "peníz". V tuto chvíli přistupujeme my s naší sociální sítí, která Vám přiloží ruku k dílu, je kompletně bezplatná a je poháněna na vynikající recenze.

Konečným výsledkem je tedy již zmiňovaná sociální síť, kde uživatel má možnost si prohlížet recepty publikované ostatními uživateli a také samozřejmě postovat svoje kulinářské výtvory (název, popis receptu seznam potřebných ingrediencí a ukázka formou fotografie). Každý přihlášený uživatel bude moci interagovat s ostatními sdílenými recepty formou liků a komentářů. Avšak nic z tohoto nebude přístupné pro neregistrovaného uživatele. Uživatel poté má možnost spravovat svůj vlastní účet s jeho recepty a taky ty, s kterými jakkoliv interagoval. Když bude doma prázdná lednička a omezený počet potravin, je možno si recepty důkladně vyfiltrovat podle obsažených ingredience či jen názvů receptů.

S touto sociální sítí může uživatel zapomenout na jakékoli doručovací služby jako Foodora, Wolt atd., s kolikrát velmi přepálenými cenami. S naší pomocí i ostatních uživatelů, si budete moci výborně uvařit, ikdyž budete mít k dispozici minimum ingrediencí i zkušeností. Není nad výborné jídlo uvařené vlastnoručně za pohodlí domova.

3. Technologie

3.1. Django

Pro napsání projektu jsme použili Django (verze 5.0.4), což je webový framework napsaný v jazyce Python. Django nabízí velkou sadu nástrojů, které usnadňují vývoj webových aplikací a umožňují tvorbu komplexních systémů s minimálním opakováním kódu. Vybrali jsme právě Django, protože používá architekturu *Model-View-Controller* (MVC), která pomáhá oddělit logiku aplikace od její prezentace. Modely reprezentují datovou vrstvu aplikace, pohledy obsahují obchodní logiku a šablony slouží k prezentaci dat uživateli.

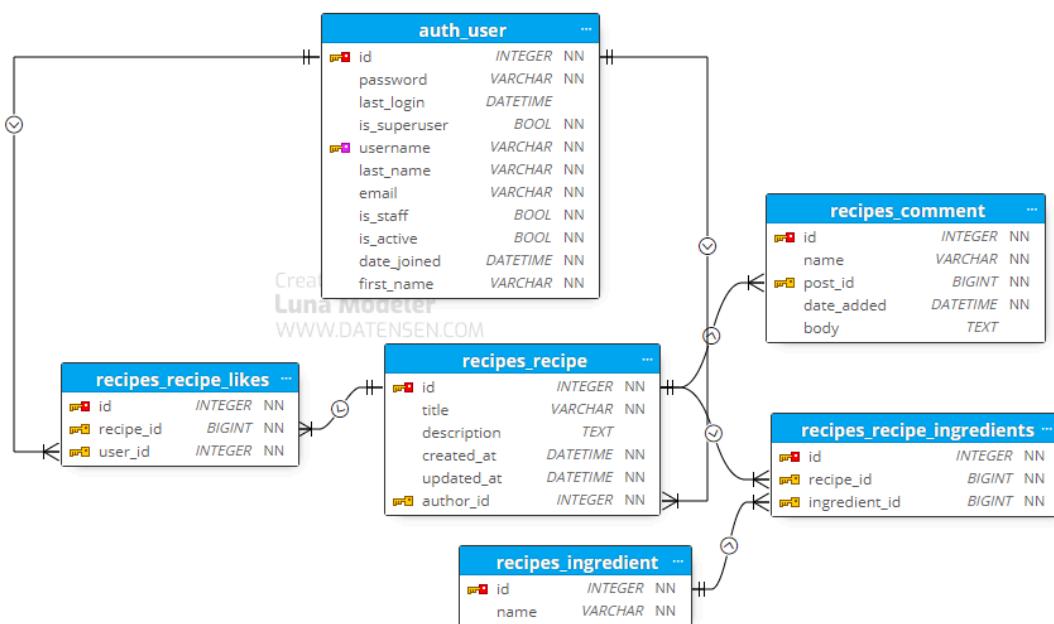
3.2. Uživatelské rozhraní

Pro náš projekt jsme se zaměřili na vytvoření moderního a responzivního uživatelského rozhraní, které by poskytovalo uživatelům příjemnou zkušenost. K dosažení tohoto cíle jsme využili následující technologie: **Bootstrap, HTML, CSS**.

Bootstrap je framework pro vývoj responzivního a mobilně přizpůsobivého webového designu. Jeho široká škála komponent a CSS stylů umožnila nam rychlý a jednoduchý vývoj uživatelského rozhraní, které je dobře strukturované a snadno použitelné. V našem projektu taky byl využit základní stavební kámen webových stránek – **HTML**, k definici struktury našich stránek, včetně značek pro obsah, odkazy, formuláře a další(/*templates*). Taky byl použit **CSS** k definici vzhledu a stylu našich stránek (/static/base.css). S pomocí CSS jsme upravili barevnou paletu, rozložení a další vizuální prvky našeho uživatelského rozhraní tak, aby odpovídaly našim požadavkům na design.

3.3. Databáze

Pro náš projekt jsme zvolili použití databáze SQLite souboru db.sqlite3. Důvodem pro tuto volbu byla kombinace jednoduchosti použití, přenositelnosti a dostačné výkonnosti pro naše potřeby. Díky vestavěné podpoře pro SQLite v Django není nutné provádět žádné složité konfigurace. Stačí jednoduše nastavit databázový soubor a Django se postará o zbytek.



Obr.1 - Diagram databáze: Vlastní

V naší aplikaci používáme několik tabulek pro ukládání různých typů dat. Zde je stručný popis každé tabulky:

auth_user: Tato tabulka obsahuje informace o uživatelských účtech. Obsahuje pole pro identifikátor uživatele (*id*), uživatelské jméno (*username*), e-mail (*email*), heslo (*password*) a další atributy.

recipes_recipe: Tabulka uchovává informace o receptech v naší aplikaci. Obsahuje pole pro identifikátor receptu (*id*), název receptu (*title*), popis (*description*), datum vytvoření (*created_at*), datum aktualizace (*updated_at*) a další atributy.

recipes_comment: Tato tabulka ukládá komentáře k receptům. Obsahuje pole pro identifikátor komentáře (*id*), jméno autora (*name*), e-mail (*email*), text komentáře (*body*) a identifikátor receptu, ke kterému komentář patří.

recipes_ingredient: Tabulka uchovává informace o ingrediencích použitých v receptech. Obsahuje pole pro identifikátor ingredience (*id*), identifikátor receptu (*recipe_id*) a identifikátor ingredience (*ingredient_id*).

recipes_recipe_likes: Tato tabulka obsahuje informace o lajcích receptů. Obsahuje pole pro identifikátor lajků (*id*), informace o uživateli (*user_id*), identifikátor receptu (*recipe_id*) a další atributy.

Každá tabulka slouží k ukládání specifických informací a je propojena s ostatními tabulkami pomocí vztahů, jako jsou cizí klíče (např. Tabulka *recipes_recipe* obsahuje sloupec *author_id*, který slouží jako cizí klíč k identifikátoru uživatele v tabulce *auth_user*.)

3.4. Dodatečné knihovny

3.4.1. Pillow

Pillow je populární knihovna pro manipulaci s obrázky v jazyce Python. Je to fork původní knihovny *Python Imaging Library* (PIL) a poskytuje širokou škálu funkcí pro načítání, úpravu a ukládání obrázků různých formátů. V kontextu webových aplikací, jako je Django, je Pillow často používán pro manipulaci s obrázky nahranými uživateli. Pillow umožňuje také snadnou konverzi mezi různými formáty obrázků, což bylo užitečné pro tvorbu naše dynamické webové stránky. [1]

3.4.2. Requests

Requests je jednoduchá knihovna pro Python, která umožňuje snadné provádění *HTTP* požadavků (metoda, kterou webové prohlížeče používají pro komunikaci se servery na internetu) jako jsou

GET, POST, PUT, DELETE a další. Knihovna poskytuje snadno použitelné metody pro odesílání dat, zpracování cookies, manipulaci s hlavičkami a další.. [2]

```
def register(request):
    if request.method == "POST":
        form = UserRegisterForm(request.POST, request.FILES)
        if form.is_valid():
            user = form.save(commit=False)
            username = form.cleaned_data.get('username')
            user.set_password(form.cleaned_data.get('password1'))
            user.save()
            profile = UserProfile.objects.get(user=user)
            profile.pfp = form.cleaned_data.get('pfp')
            profile.save()
            messages.success(request, f"{username}, your account is created, please login.")
            return redirect('user-login')
    else:
        form = UserRegisterForm()
    return render(request, 'users/register.html', {'form': form})
```

Obr.2 - Ukázka kódu (metoda POST) :Vlastní

V tomto kódu se využívá metoda *HTTP* požadavku, konkrétně *POST*. Tato podmínka zjišťuje, zda byl *HTTP* požadavek odeslán metodou *POST*. Metoda *POST* se používá obvykle při odesílání formulářů, když uživatel zadává nebo upravuje data na stránce.

3.4.3. Django Crispy Forms a Crispy-Bootstrap4

Taky jsme použili knihovnu Django Crispy Forms(verze 2.1), která umožňuje snadnou integraci a formátování formulářů ve webových aplikacích. Tato knihovna poskytuje jednoduché řešení pro vytváření a úpravu formulářů pomocí konfigurovatelných šablon a stylů.

K tomu jsme použili knihovnu Crispy-Bootstrap4(verze 2024.1), což je adaptér pro knihovnu Django Crispy Forms, který umožňuje jednoduchou integraci s frameworkm Bootstrap 4 a využití jeho funkcí pro responzivní design a pokročilé komponenty formulářů. [3] [4]

3.4.4. Django-ckeditor a Django-richtextfield

Django-ckeditor je modul pro Django, který poskytuje integrovaný editor *WYS/WYG* (What You See Is What You Get) pro správu obsahu ve webových aplikacích. Tato knihovna je založena na populárním editoru CKEditor a umožňuje uživatelům jednoduše vytvářet a upravovat obsah prostřednictvím bohatého uživatelského rozhraní. V našem projektu jsme se rozhodli využít Django-ckeditor(verze 6.7.1) pro implementaci *RichTextField*, díky integraci s knihovnou Django-richtextfield(verze 1.6.2), která využívá CKEditor jako základní editor pro manipulaci s textovým obsahem. RichTextField je textové pole s bohatým formátováním obsahu. Tato funkcionality umožňuje uživatelům snadno vytvářet obsah s formátováním jako v běžných textových editorech. [5] [6]

3.4.5. Django-modeltranslation

Použili jsme knihovnu Django-modeltranslation(verze 0.18.11), která umožňuje snadnou lokalizaci dat uložených v modelu, ale knihovnu jsme využili primárně pro internacionalizaci HTML šablon. To umožňuje vytvářet vícejazyčné webové aplikace bez zbytečné komplikace. My jsme ji použili pro překlad naše aplikace z češtiny do angličtiny a naopak. [7]

4. Scény aplikace

4.1. Registrace

Pro registraci nového uživatele do systému jsme vytvořili formulář (viz. obr.3). Obsahuje pole Username (uživatelské jméno), Password1 (heslo), Password2 (potvrzení hesla) a Pfp (profilové foto) s možností nahrát soubor. Taky tlačítko Sign Up (registrovat se) a odkaz na přihlašování, pokud uživatel už má vytvořený účet.

The screenshot shows a registration form titled "Registrovat se!" (Register) on a background decorated with various food and drink icons. The form contains the following fields:

- Username*
- Password1*
- Password2*
- Confirm Password
- Pfp
- Vybrat soubor (Select file)
- Soubor nevybrán (File not selected)
- Sign Up (highlighted in orange)

Below the form, there is a link: "Jáž máte účet? Přihlaste se." (Already have an account? Log in.)

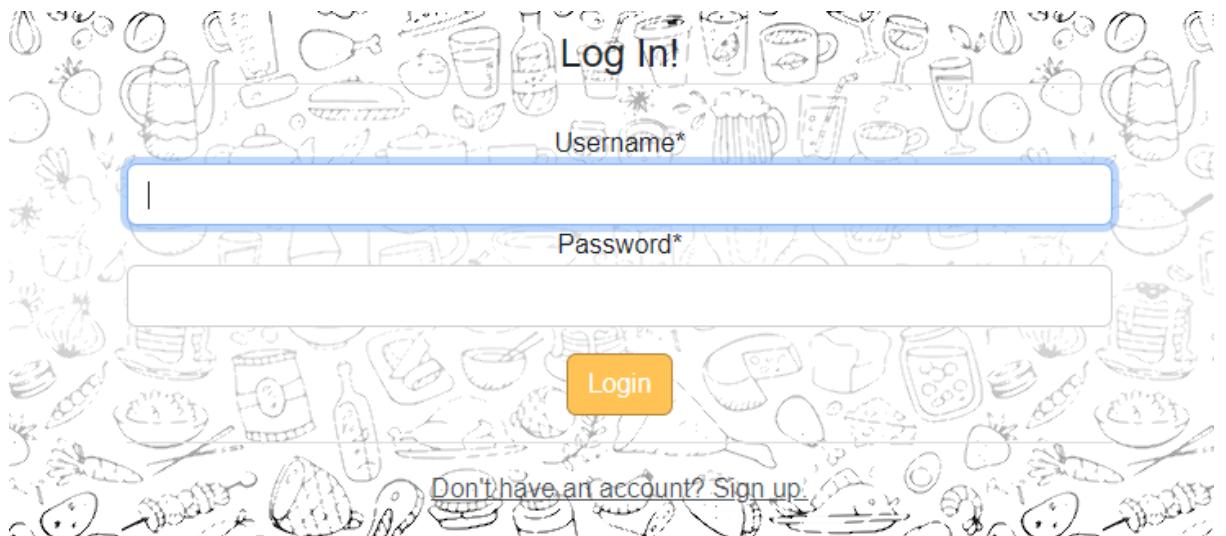
Obr.3 - Ukázka scény "Registrace"

Pro zpracovávání registrace používáme funkci `register(request)`. Nejprve funkce zkонтroluje, zda byl požadavek odeslán metodou `POST`(viz. kapitola 3.4.2). Pokud je požadavek typu `POST`, funkce vytvoří instanci formuláře `UserRegisterForm` a předá jí data z požadavku pomocí `request.POST` a `request.FILES`. Poté funkce zkонтroluje, zda jsou data vložená do formuláře platná. To je provedeno pomocí metody `is_valid()`. Pokud jsou data formuláře platná, funkce uloží informace o nově registrovaném uživateli do databáze. Pokud byl v registračním formuláři nahrán profilový obrázek, funkce ho uloží do databáze jako profilovou fotografii uživatele. Nakonec funkce přesměruje uživatele na stránku pro přihlášení pomocí funkce `redirect('user-login')`.

4.2. Přihlášení

Pro přihlášení uživatele do systému jsme vytvořili formulář(viz. obr.4).

Obsahuje pole Username (uživatelské jméno) a Password (heslo). Taky tlačítko Login (přihlásit se) a odkaz na registraci, pokud uživatel není zaregistrovaný.

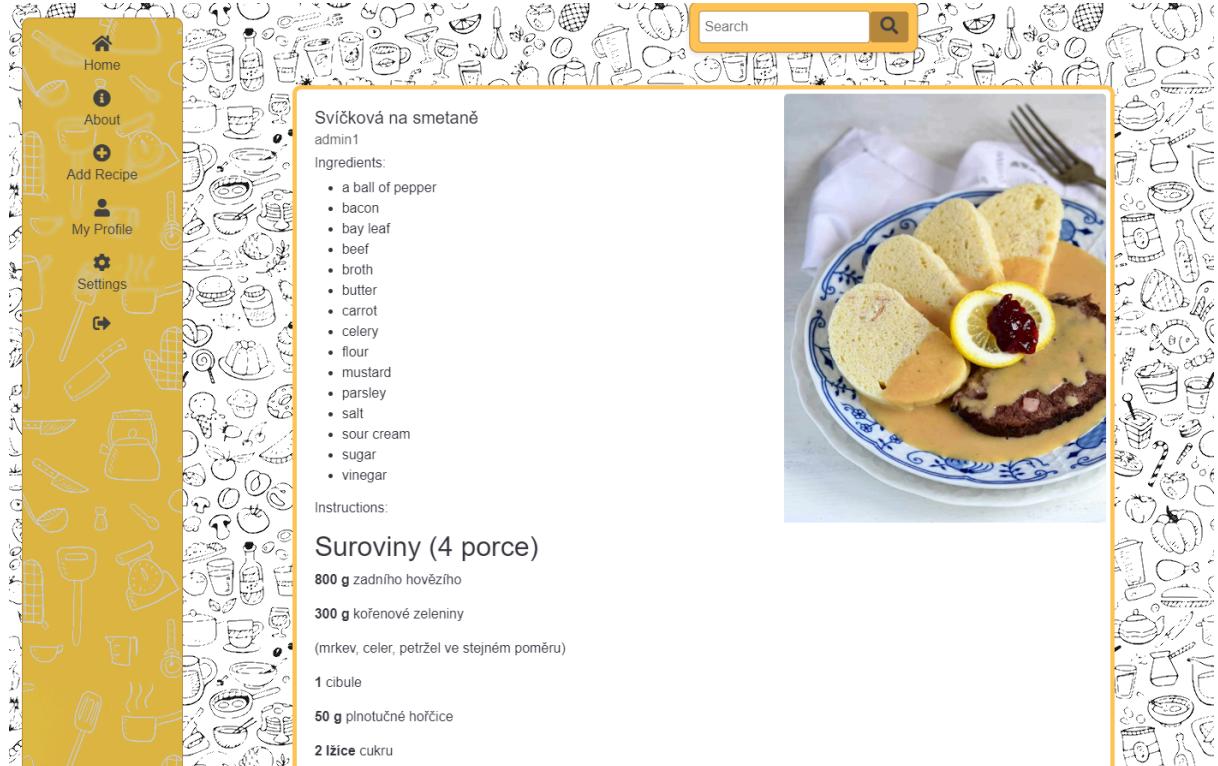


Obr.4 - Ukázka scény “Přihlašování”

Funkčnost přihlašování je implicitně implementována pomocí dekorátoru `@login_required`, který se používá ve funkcích jako `profile`, `settings` a `change_password`. Tento dekorátor zajišťuje, že uživatel musí být přihlášen, aby mohl přistupovat k těmto funkcím. Pokud se uživatel pokusí přistoupit k jedné z těchto funkcí, aniž by byl přihlášen, Django automaticky přesměruje na stránku přihlášení. Samotný proces přihlášení, kdy uživatel zadá své přihlašovací údaje a autentizuje se, je řešen pomocí vestavěných autentizačních pohledů Djanga.

4.3. Domovská stránka

Pro zobrazení receptů všech uživatelů jsme vytvořili domovskou stránku (viz. obr.5). Tady můžeme vidět recepty, nahoře Search Bar (vyhledávání) a taky Navbar (navigační panel) vlevo.



Obr.5 - Ukázka scény "Domovská stránka"

Pro zobrazení domovské stránky uživatelům používáme funkci `home(request)`. Ve funkci se nejprve získají všechny recepty z databáze pomocí modelu `Recipe`. Tyto recepty jsou seřazeny podle data vytvoření (`created_at`) v sestupném pořadí. Následně je vytvořen slovník `context`, který obsahuje seznam receptů pod klíčem '`recipes`'. Nakonec je volána funkce `render()`.

4.4. O nás

Tady jsme vytvořili šablonu (viz. obr.6), která slouží k uvítání uživatele na sociální síti receptů, představuje klíčové vlastnosti této sociální sítě a seznamuje uživatele s týmem za projektem.



Obr.6 - Ukázka scény "O nás"

Funkce, zodpovědná za zobrazení stránky "O nás" uživatelům je `about(request)`. Tělo funkce jednoduše vrací výsledek funkce `render()`.

4.5. Vytváření receptů

Pro vytvoření nového recepta jsme udělali formulář (viz. obr.7), který obsahuje Title (název), Description (popis), Ingredients (seznam ingrediencí) a Photo (foto).

The screenshot shows a web-based form for creating a new recipe. At the top, there's a title bar with icons for file operations like back, forward, and search. Below it, the main title is 'Přidat Recept'. The form fields include:

- A 'Název:' (Name) input field.
- A 'Popis:' (Description) input field.
- A toolbar with various text styling options (bold, italic, underline, etc.) and a 'Zdroj' (Source) button.
- An 'Ingredients:' section containing a list of ingredients with checkboxes. The list includes: apple, avocado, bacon, beans, beef, beef stock, bell pepper, blueberries, brie, broccoli, butter, canned beans, canola oil, caramel, carrot, celery, cheddar, chicken, chicken stock, chocolate, corn, cottage cheese, cranberry, cucumber, feta, fish, flour, garlic, ginger, gouda, green peas, ham, honey, ketchup, kiwi, lemon, lettuce, mango, marjoram, mascarpone, mayonnaise, meat, milk, mozzarella, muesli, mushroom, mustard, oats, olive, olive oil, onions, orange, oregano, paprika, parmesan, pasta, pear, pepper, pork, potatoes, pumpkin, red chilli, red wine, rice, rice noodles, ricotta, rosemary, salmon, salt, and sausage.

Obr.7 - Ukázka scény "Vytváření receptů"

Třída, určena pro vytváření nových záznamů receptů v databázi je *RecipeCreateView*. Třída dědí z třídy *LoginRequiredMixin*, což znamená, že uživatel musí být přihlášen, aby mohl mít přístup k tomuto pohledu. Formulář, který se použije pro zadávání dat, je určen třídou *RecipeForm*. Název šablony, která bude použita pro vykreslení tohoto pohledu, je specifikován jako '*recipes/recipe_form.html*'.

Metoda *form_valid(self, form)* je přepsána, aby zpracovala odeslaný formulář. Nejprve se zavolá metoda *form.save(commit=False)*, aby byl vytvořen nový objekt receptu, ale nebyl ještě uložen do databáze.

Na ingredience jsme rozhodli použít třídu, *IngredientCheckboxSelectMultiple* (viz. obr.8), což je vlastní widget pro pole formuláře. Je tam podtřída *forms.CheckboxSelectMultiple*, což je widget, který vykresluje seznam zaškrťvacích polí.

```
class IngredientCheckboxSelectMultiple(forms.CheckboxSelectMultiple):
    def render(self, name, value, attrs=None, renderer=None):
        if value is None:
            value = []
        final_attrs = self.get_context(name, value, attrs)['widget']['attrs']
        output = [mark_safe('<div class="column">')]
        str_values = {str(v) for v in value} # Normalize to strings.
        for i, (option_value, option_label) in enumerate(self.choices):
            if i % 10 == 0 and i != 0:
                output.append(mark_safe('</div><div class="column">'))
            id_ = final_attrs.get('id') + '_{}'.format(i) # Construct id for each checkbox
            cb = forms.CheckboxInput(final_attrs, check_test=lambda value: value in str_values)
            rendered_cb = cb.render(name, option_value)
            output.append('<label for="{}">{}</label>'.format(id_, rendered_cb + ' ' + option_label))
        output.append(mark_safe('</div>'))
        return mark_safe('\n'.join(output))
```

Obr.8 - Ukázka kódu (class *IngredientCheckboxSelectMultiple*): Vlastní

forms.CheckboxSelectMultiple je standardní widget Django pro výběr více položek zaškrťvacími polí. Přepisuje metodu render pro úpravu způsobu, jakým jsou zaškrťvací políčka vykreslena. Seskupuje zaškrťvací políčka do sloupců, každý sloupec obsahuje maximálně 10 zaškrťvacích políček. Přidává štítky ke každému zaškrťvacímu políčku, které jsou spojeny s vstupy zaškrťvacích políček. Štítky jsou generovány na základě poskytnutých možností. Pro každé zaškrťvací políčko konstruuje jedinečné ID, aby se zajistilo, že jsou správně spojeny se svými štítky. Normalizuje hodnoty na řetězce, aby se zajistila konzistence. Nakonec vrátí vykreslené HTML jako řetězec, používá *mark_safe* k označení jako bezpečného.

4.6. Detaily receptů

Pro zobrazení detailů recepta jsme vytvořili šablonu(viz. obr.8).

Obsahuje dvě tlačítka Edit (upravit) a Delete (smazat), taky vidíme tlačítko Like (líbí se) a komentáře s odkazem na vytvoření nového.

The screenshot shows a detailed view of a recipe. On the left, there's a vertical sidebar with icons related to cooking (spoon, bowl, etc.). The main content area has a title "Postup přípravy" (Preparation steps) with five numbered steps. Below the steps is a date "duben 26, 2024". Underneath the date are two buttons: "Upravit" (Edit) and "Smazat" (Delete). A "Líbí se" button with "0 Líbí se" is shown. The section "Komentáře" (Comments) contains the message "Zatím Žádné Komentáře..." and a link "Přidej jeden".

Postup přípravy

- Maso prošpikujeme hranolky slaniny (jde to skvěle, pokud ji dáme na půl hodiny do mrazničky), osolíme a opečeme na oleji, nebo na oleji s máslem (máslo se nebude připalovat).
- Poté maso vyjmeme a na výpeku opečeme nahrubo nastrouhanou zeleninu a cibuli. Jakmile mrkev pustí barvu, přidáme cukr a necháme zkaramelizovat.
- Přidáme hořčici, koření, přilijeme ocet, vložíme maso a zalijeme vroucím vývarem. Vložíme hrnec či pekáč i s poklicí do trouby a pečeme do změknutí masa (přibližně 1,5-2 hodiny ve středně vyhřáté troubě). Poté maso i koření vyjmeme, maso nakrájíme a omáčku rozmixujeme.
- Je-li třeba, zahustíme omáčku hladkou moukou, rozkvedlanou v mléce a provaříme (mouku je třeba provářit minimálně po dobu 20 minut), podle potřeby dosolíme.
- Omáčku zjemníme smetanou, dochutíme citronem a plátkem másla, již nevaříme. Podáváme s houskovým knedlíkem, na talíři dozdobíme kouskem citronu a brusinkami.

duben 26, 2024

Upravit Smazat

Líbí se - 0 Líbí se

Komentáře

Zatím Žádné Komentáře...
[Přidej jeden](#)

Obr.9 - Ukázka scény “Detaily receptů”

Třída, která slouží k zobrazení detailů konkrétního receptu je *RecipeDetailView*. Třída dědí z třídy *DetailView*, která je součástí frameworku Django a poskytuje obecný pohled pro zobrazení detailů objektu. Název šablony, která bude použita pro vykreslení tohoto pohledu, je specifikován jako '*recipes/recipe_detail.html*'.

Metoda *get_context_data(self, *args, **kwargs)* je přepsána tak, aby přidala další kontextová data do šablony pro vykreslení. Nejprve

získává objekt receptu na základě předaného identifikátoru (pk) z URL pomocí funkce `get_object_or_404()`.

Následně vypočítává celkový počet "lajků" (likes) pro tento recept pomocí metody `total_likes()`, která je definována v modelu Recipe.

Také určuje, zda je aktuální uživatel přihlášen a zda již tento recept označil jako oblíbený. To se provádí pomocí kontroly, zda existuje záznam o lajku pro aktuálního uživatele v poli likes receptu.

4.7. Komentáře

Pro napsání komentáře jsme vytvořili formulář(viz. obr.10), který obsahuje jenom Body (tělo).

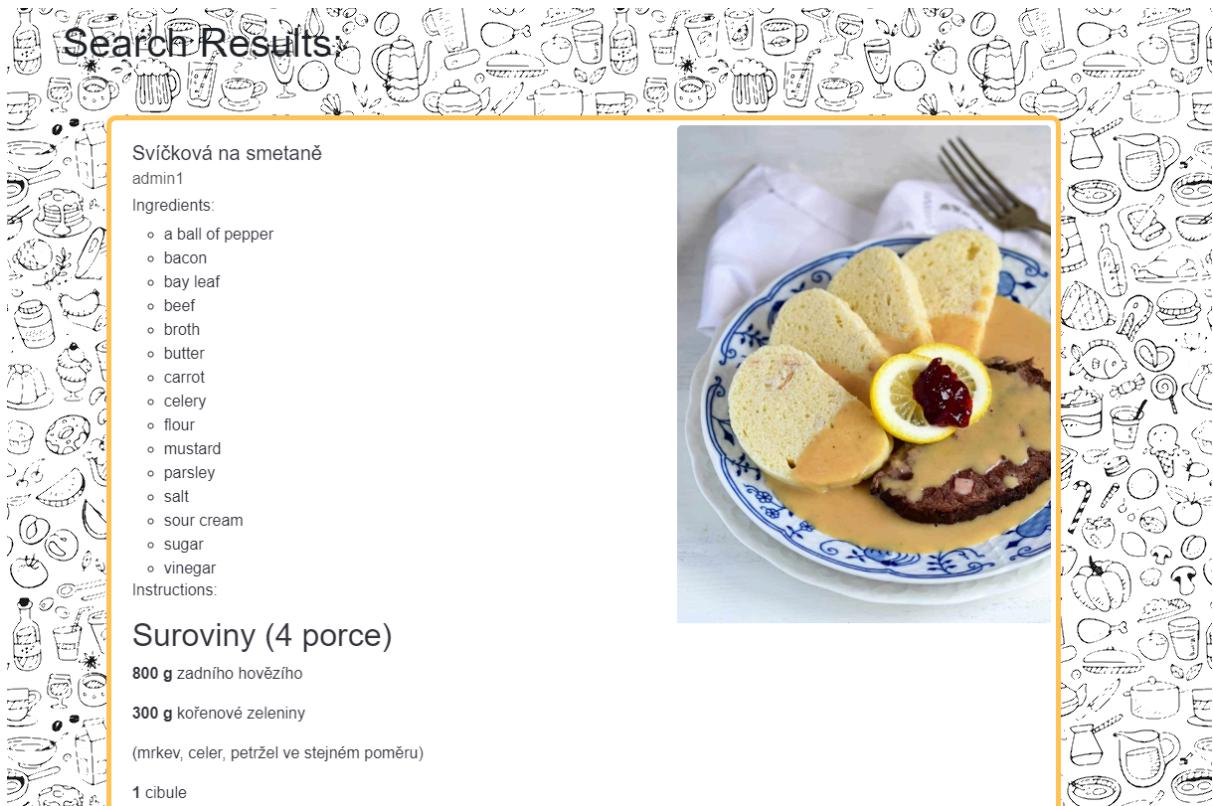
The screenshot shows a web-based form for adding a comment. The title of the form is "Add Comment...". Below the title is a label "Body:" followed by a text area. Above this text area is a toolbar with various formatting options like bold (B), italic (I), underline (U), strikethrough (S), and a link icon (L). The text area has a placeholder "Type your comment here". At the bottom of the form is a dark blue button with the text "Add Comment" in white.

Obr.10 - Ukázka scény "Přidání komentáře"

Používáme třídu `AddCommentView` k zobrazení formuláře pro přidání komentáře k receptu a zpracování a uložení dat tohoto formuláře do databáze.

4.8. Vyhledávání receptů

Po zadání ingredience do Search (vyhledávání) se zobrazí stránka Search Results (výsledek vyhledávání) s receptem, který obsahuje dané ingredience (viz. obr.10).



Obr.11 - Ukázka scény "Výsledek vyhledávání"

Funkce `search_recipes` umožňuje uživatelům vyhledávat recepty podle zadaných ingrediencí. Funkce `search_recipes` je volána, když uživatel odešle formulář pro vyhledávání receptů.

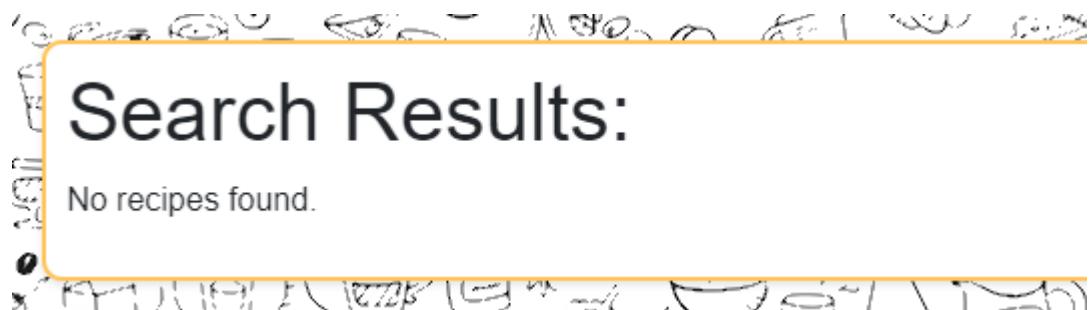
Vytváří instanci formuláře `RecipeSearch` s daty získanými z požadavku (GET data). To umožňuje získat data z formuláře, který uživatel vyplnil.

`ingredients = form.cleaned_data['ingredients']` získá seznam ingrediencí z čistých dat formuláře. `recipes = models.Recipe.objects.filter(ingredients__name__in=ingredients.split(','))` provede dotaz na databázi, aby našel recepty obsahující všechny zadané ingredience.

Tento dotaz používá ORM Django a operátor filter, aby vybral recepty,

jejichž ingredience mají názvy obsažené v seznamu zadaných ingrediencí.

V případě, že při zadání ingredience se nenašel žádný recept, tak se zobrazí stejná stránka(viz. obr.11). Obsahuje ale jen text *No recipes found* (nebyly nalezené žádné recepty).



Obr.12 - Ukázka scény “Výsledek vyhledávání - žádné recepty”

4.9. Profil

Profilová stránka(viz. obr.12) uživatelé obsahuje pfp (profilové foto), Username (uživatelské jméno), tlačítko Liked recipes (oblíbené recepty) a recepty, které byly vytvořené uživatelem.

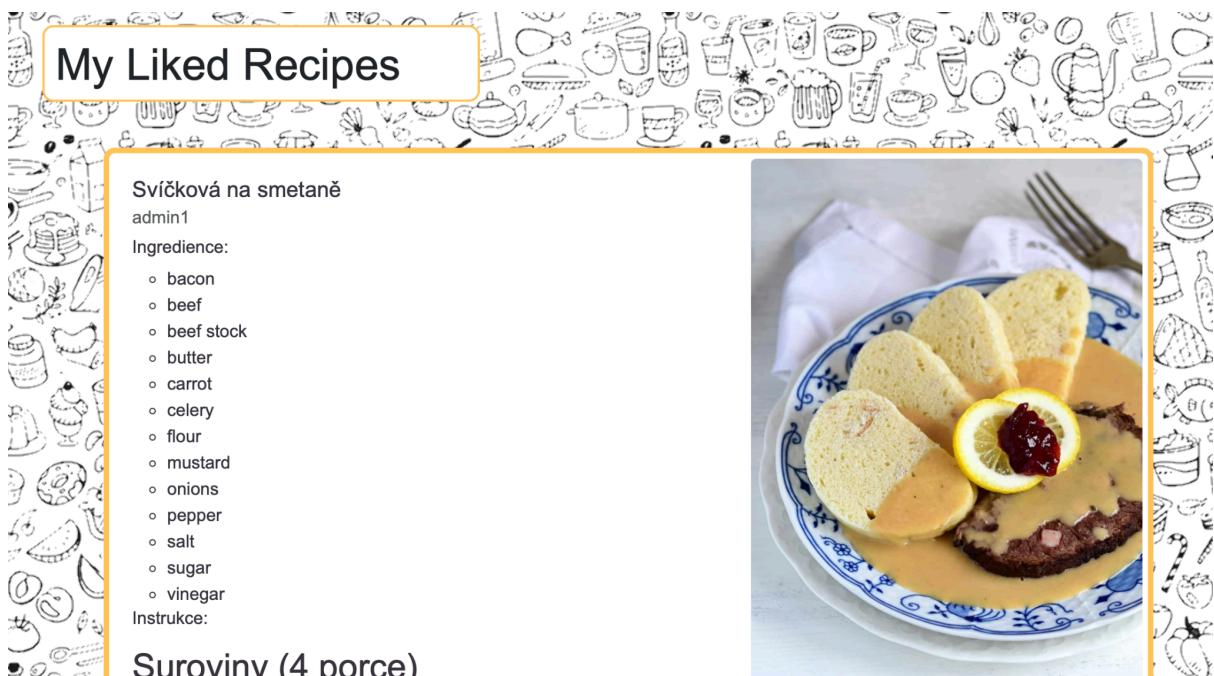
A screenshot of a user profile page. At the top left is a placeholder for a profile picture. Next to it is the username 'admin1'. Below the username is a button labeled 'Tvoje recepty:' and another button labeled 'Liked recipes'. The main content area shows a recipe card for 'Svíčková na smetaně' (Beef tenderloin in cream sauce). The card includes the author's name ('admin1'), a list of ingredients ('bacon, beef, beef stock, butter, carrot, celery, flour, mustard, onions, pepper, salt, sugar, vinegar'), and cooking instructions. To the right of the card is a photograph of a meal served on a blue and white plate, consisting of a piece of meat, a slice of bread, and a lemon wedge.

Obr.13 - Ukázka scény “Profil”

Funkce `profile` získává aktuálního uživatele pomocí `request.user`, poté pomocí tohoto uživatele filtruje recepty, které jsou přiděleny tomuto uživateli. Nakonec vrací odpověď, ve které vykresluje šablonu '`uzivatele/profil.html`' s předaným kontextem obsahujícím seznam receptů přidělených uživateli.

4.10. Oblíbené recepty

Pro zobrazení oblíbených receptů uživatelé, jsme vytvořili stránku My Liked Recipes(moje oblíbené recepty). Obsahuje recepty, kterým uživatel dal "like".(viz. obr.13)



Obr.14 - Ukázka scény "Moje oblíbené recepty"

Funkce `liked_recipes` získává všechny recepty, které byly uživatelem označeny jako oblíbené. K tomu používá filtraci na základě pole `likes` u modelu `Recept`, kde `request.user` je uživatel, jenž má být ve filtru.

4.11. Nastavení

Stránka Nastavení (viz. obr.14) obsahuje Username (uživatelské jméno), pfp (profilové foto) a tlačítko Update (aktualizace), které uživatel může případně změnit, taky má tlačítko Change Password (změna hesla).

Settings

Username*

admin1

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Pfp

Currently: [pfp/default.png](#) Clear

Change: Выберите файл Файл не выбран

Update

Change Password

Obr.15 - Ukázka scény “Nastavení”

Funkce `settings` začíná kontrolou, zda byl požadavek zaslán metodou `POST`. Pokud je metoda požadavku `POST`, funkce zpracovává data formulářů pro aktualizaci profilu. Pokud jsou formuláře platné, provede aktualizaci dat profilu a zobrazí zprávu o úspěšné aktualizaci. Poté uživatele přesměruje na jejich profilovou stránku. Pokud je metoda požadavku `GET`, funkce získává instance formulářů pro aktualizaci profilu a vykreslí stránku pro nastavení s těmito formuláři.

4.12. Změna hesla

Pro změnu hesla jsme udělali stránku(viz. obr.16), která obsahuje pole Old password (staré heslo), New password (nové heslo), New password confirmation (potvrzení nového hesla) a tlačítko Change Password (změnit heslo).

The screenshot shows a web form titled "Change Password". The form contains four input fields: "Old password*", "New password*", "New password confirmation*", and a "Change Password" button. The "Old password*" field is highlighted with a blue border. Below the "New password*" field is a list of validation rules:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Obr.16 - Ukázka scény „Změna hesla“

Žádosti o změnu hesla uživatele zpracovává funkce `change_password`. Funkce začíná kontrolou, zda byla žádost zaslána metodou `POST`. Pokud byla žádost zaslána metodou `POST`, funkce vytvoří instanci formuláře `PasswordChangeForm`, která obdrží aktuálního uživatele (`request.user`) a data z požadavku (`request.POST`). Po úspěšné změně hesla se aktualizuje aktuální relace uživatele voláním funkce `update_session_auth_hash()`.

5. Instalace, setup a spuštění

Správná instalace a nastavení projektu z vašeho repozitáře. Prvním krokem je potřeba si naklonovat repozitář do svých souborů. Toho se dosáhne otevřením příkazového řádku a spuštěním příkazu **git clone <<https://github.com/gyarab/2023-3e-Zeryk>>**.

Jakmile je repozitář naklonovaný, je potřeba přejít do složky s projektem a vytvořit virtuální prostředí pomocí příkazu **python -m venv venv**. Následně musíme aktivovat náš venv příkazem **source ./venv/Scripts/activate**.

Nyní se musí nainstalovat všechny potřebné závislosti. Nejprve spustit příkaz **pip install -r requirements.txt**, který nainstaluje všechny balíčky uvedené v souboru requirements.txt.

Po instalaci závislostí se bude muset provést migrace. Toho se dosáhne spuštěním příkazu **python manage.py migrate**. Tento příkaz zajistí, že databáze bude připravena pro běh aplikace.

Nakonec už jenom stačí spustit vývojový server. Stačí zadat příkaz **python manage.py runserver**, a to spustí server na lokálním počítači. Poté se stačí připojit k aplikaci ve webovém prohlížeči pomocí adresy **http://127.0.0.1:8000/en**.

6. Závěr

Ke konci bychom rádi shrnuli naši práci na tomto projektu po dobu tohoto školního roku. Shodli jsme se na tom, že to bylo vskutku náročné, ať už ze stránky technické, tak ze stránky komunikační. Ne vždy šlo vše hladce podle plánů a očekávání, ale i přesto jsme byli schopni naše zadání pokořit a splnit. Výsledkem je plně funkční sociální síť pro sdílení receptů, na kterou jsme právem pyšní. Pozitivem této ročníkové práce bylo si vyzkoušet pracovat v týmu na nějakém větším a náročnějším projektu. Každý z nás si teď může dostatečně promyslet, která z variant zpracování ročníkového projektu se mu více zamlouvala (týmová či samostatná). Tento projekt pro každého z nás vyvolával velikou výzvu, kterou jsme byli odhodlání zdolat a vést tak naše programátorské schopnosti v oblasti IT k lepšímu. Objevovali jsme a zkoušeli nové techniky, které doufáme, že nám v budoucnu ještě párkrt poslouží.

7. Zdroje

- [1] Pillow. Python Imaging Library. Clark. Apr 1, 2024
<https://pypi.org/project/pillow/>.....5
- [2] Co je to HTTP request | DAMI
<https://www.damidev.com/slovnik/http-request>.....6
- [3] Forms have never been this crispy.
<https://django-crispy-forms.readthedocs.io/en/latest/>.....6
- [4] crispy-bootstrap4 2024.1. Smith. Feb 27, 2024
<https://pypi.org/project/crispy-bootstrap4/>.....6
- [5] Django admin CKEditor integration.
<https://django-ckeditor.readthedocs.io/en/latest/>.....7
- [6] django-richtextfield 1.6.2. Roes. Mar 7, 2024
<https://pypi.org/project/django-richtextfield/>.....7
- [7] Modeltranslation.
<https://django-modeltranslation.readthedocs.io/en/latest/>....7

8. Zdroje využité při programování

[1] <https://www.w3schools.com/css/>

(backgrounds, colors, border, margin, padding, dimension, box model, outline, text, max_width, align, navbar, dropdowns, image_gallery, border_images, buttons, user_interface)

[2] (Like button)

<https://www.youtube.com/watch?v=PXqRPqDjDgc&list=PLCC34OHNcOtr025c1kHSPrnP18YPB-NFi&index=18>

[3] (Comment posting)

https://www.youtube.com/watch?v=OuOB9ADT_bo&list=PLCC34OHNcOtr025c1kHSPrnP18YPB-NFi&index=35

[4] (Blog comment section)

<https://www.youtube.com/watch?v=hZrlh4qU4eQ&list=PLCC34OHNcOtr025c1kHSPrnP18YPB-NFi&index=34>

[5] (RichTextField to Django blog)

<https://www.youtube.com/watch?v=mF5jzSXb1dc&list=PLCC34OHNcOtr025c1kHSPrnP18YPB-NFi&index=21>

[6] (Basic HTML and CSS)

<https://www.youtube.com/watch?v=iG2jotQo9NI>

[7] (Python Django Web)

<https://www.youtube.com/watch?v=F5mRW0jo-U4>

9. Seznam Obrázků

1	Diagram databáze (vlastní)	4
2	Ukázka kódu (metoda POST)	6
3	Ukázka scény “Registrace”.....	8
4	Ukázka scény “Přihlašování”.....	9
5	Ukázka scény “Domovská stránka”.....	10
6	Ukázka scény “O nás”.....	11
7	Ukázka scény “Vytváření receptů”.....	12
8	Ukázka kódu (class IngredientCheckboxSelectMultiple)....	13
9	Ukázka scény “Detaily receptů”	14
10	Ukázka scény “Přidání komentáře”.....	15
11	Ukázka scény “Výsledek vyhledávání”.....	16
12	Ukázka scény “Výsledek vyhledávání - žádné recepty”.....	17
13	Ukázka scény “Profil”.....	17
14	Ukázka scény “Moje oblíbené recepty”.....	18
15	Ukázka scény “Nastavení”.....	19
16	Ukázka scény “Změna hesla”.....	20