



Документација за тимски проект RecipeAI

Верзија 1.0

Датум: 21.03.2024

| | |
|--|----------|
| 1. Вовед..... | 3 |
| 2. Клучни функции на апликацијата..... | 3 |
| 2.1. Автентикација на корисници..... | 3 |
| 2.2. Управување со рецепти..... | 3 |
| 2.3. Откривање на нови рецепти..... | 3 |
| 2.4. Персонализација..... | 3 |
| 2.5. Интеракција со заедницата..... | 4 |
| 3. Користени технологии..... | 4 |
| 3.1. Django..... | 4 |
| 3.2. HTML/CSS/JavaScript..... | 4 |
| 3.3. Bootstrap..... | 4 |
| 3.4. SQLite..... | 4 |
| 4. Модели во апликацијата..... | 4 |
| 5. Views во апликацијата..... | 5 |
| 6. URLs во апликацијата..... | 6 |
| 7. Templates во апликацијата..... | 7 |
| 8. Forms во апликацијата..... | 8 |

1. Вовед

Во овој документ е обезбеден комплетен преглед на архитектурата, функционалноста и деталите за имплементација на нашата веб апликација, RecipeAI. RecipeAI е дизајнирана за да им обезбеди на корисниците платформа за истражување, споделување и откривање кулинарски креации. Без разлика дали корисниците бараат инспирација, сакаат да допринесат со свои рецепти или да се вклучат во заедницата на истомислени ентузијастички за храна, нашата апликација нуди многу функции што одговараат на нивните потреби.

2. Клучни функции на апликацијата

2.1. Автентикација на корисници

- Корисниците можат да креираат своја сметка, да се најават и одјават безбедно. Автентикација на корисници е имплементирана со вградениот систем за автентикација на Django, обезбедувајќи приватност и безбедност на податоците.

2.2. Управување со рецепти

- Регистрираните корисници можат да додаваат, како и да ги уредуваат и бришат своите рецепти. При креирање на рецепт има најразлични податоци коишто можат да бидат додадени како што се името, време на подготовка, број на порции, состојки, категорија во која спаѓа рецептот и слично.

2.3. Откривање на нови рецепти

- При приказ на рецептите, за секој посебно може да се отворат детали за избраниот рецепт и притоа да се видат повеќе информации како и рецензии за него.

2.4. Персонализација

- Секцијата “For you” вклучува интерфејс на паметен Chatbot кој препорачува персонализирани рецепти базирани на состојки и преференци на корисникот. Односно самиот Chatbot користи RAG пристап и вектор база, трениран е на големо множество на рецепти, односно овој Chatbot е персонализиран за рецепти, каде што корисниците може да пишуваат со какви состојки располагаат и самиот Chatbot да им даде некоја идеја. Го користи GPT4 моделот “Mistral Open Orca” од 8GB. Вектор база користи

“Chroma”. Температурата на моделот е 1, со што му се дозволува на моделот да биде малку креативен, а сепак да нема многу халуцинации.

2.5. Интеракција со заедницата

- Корисниците можат да се вклучат во заедницата преку оставање на коментари и рецензии за рецепти споделени од другите корисници. Со ова е олеснета соработката и размената на мислења помеѓу корисниците.

3. Користени технологии

3.1. Django

- Нашата веб апликација е направена со помош на Django, со што се обезбедува стабилна основа за равој на скалабилни и одржливи веб-апликации.

3.2. HTML/CSS/JavaScript

- Развојот на frontend се олеснува со користење на стандардни веб-технологии за креирање на динамични и респонзивни интерфејси.

3.3. Bootstrap

- Рамката Bootstrap се користи за стилизирање и распоред на frontend-от, обезбедувајќи конзистентно и визуелно атрактивно корисничко искуство на различни уреди.

3.4. SQLite

- Користењето на SQLite за управување со бази на податоци во нашата апликација нуди практични предности во развојот и тестирањето на апликацијата. SQLite е лесен за интеграција, што значи дека не се потребни дополнителни конфигурации или поставувања за да се користи.

4. Модели во апликацијата

- Процесот на развој го започнавме со внимателна анализа на барања за нашата веб-апликација. За приказ на структурата на податоците и врските помеѓу нив, направивме ER-дијаграм, па според ова ги дефиниравме

моделите во апликацијата. Секој модел беше претставен со цел да го претстави соодветниот ентитет во ER-дијаграмот.

- Приоритет беше даден на интегритетот на податоци и флексибилноста, обезбедувајќи дека моделите можат да ги прифатат различните типови на рецепти, состојки, корисници и интеракции. Преку итеративно усовршување и соработка, го финализиравме моделот на структурата, наоѓајќи баланс помеѓу едноставноста и комплетноста за да ги поддржи функционалностите на апликацијата ефективно.

```
class CustomUser(AbstractUser):
    ROLE_CHOICES = (
        ('a', 'Admin'),
        ('u', 'User')
    )
    role = models.CharField(max_length=10, choices=ROLE_CHOICES, )
```

- Овој модел наследува од AbstractUser-от на Django, со што ги наследува сите полиња и функционалности. Исто така во CustomUser има уште едно поле, role, со што се бира улогат за корисникот.
- Покрај овој модел, ги дефиниравме и следните: Category, Ingredient, Recipe, RecipeIngredient, Instruction, Review.
- Со моделот Recipe е претставен рецепт, кој може да има име, корисник, категорија, опис, време на подготовка, време на готвење, број на сервирања и слика. Има надворешни клучеви кон CustomUser и Category.
- Со моделот RecipeIngredient е претставена врската помеѓу рецептот и состојките, вклучувајќи ја и количината и единицата за секој рецепт.
- Останатите модели се поедноставни, а во некои од нив има надворешни клучеви кон другите модели.

5. Views во апликацијата

- Django views се Python функции коишто земаат http барања и враќаат http одговори.
- contact_form_submission(): Ова view се справува со поднесување контакт форма. Прифаќа POST барања што содржат податоци за е-пошта и пораки, испраќа е-пошта со детали за поднесување формулари и враќа JSON одговор кој покажува успех или неуспех.
- index(): Се рендерира homepage на апликацијата, односно се прикажува селекција на рецепти. Ги презема најновите 6 рецепти од базата на податоци и ги пренесува на темплејтот за рендерирање.

- `user_login()`: Се справува со функционалноста за најава на корисник. Прифаќа POST барања кои содржат `credentials` за најава, го автентифицира корисникот и го најавува доколку сè е валидно.
- `register_customer()`: Ја прикажува страницата за регистрација на корисник и се справува со регистрацијата на корисникот. Ги прифаќа POST барањата кои содржат податоци за регистрација, ја потврдува формата и создава нова корисничка сметка доколку податоците се валидни.
- `logout_view()`: Го одјавува моментално најавениот корисник и го редириктира на почетната страница.
- `all_recipes()`: Рендерира страница која ги прикажува сите рецепти достапни во базата на податоци.
- `details()`: Рендерира страница со детални информации за одреден рецепт. Показува детали за рецептот, категорија и `reviews` од базата на податоци и ги прикажува во шаблонот. Исто така се справува со поднесување на нови `reviews`.
- `delete_recipe()`: Брише рецепт од базата на податоци врз основа на даденото ID.
- `add_recipe()`: Рендерира форма за додавање нов рецепт и се справува со поднесувањето на формата. Создава нов објект рецепт со дадените податоци и го зачувува во базата на податоци.
- `edit_recipe()`: Рендерира форма за уредување на постоечки рецепт и се справува со поднесувањето на формата. Го враќа објектот на рецепт што треба да се уредува, ги пополнува полињата на формата со неговите податоци и го ажурира објектот на рецепт со новите податоци по поднесувањето.
- `for_you()`: Рендерира чатбот и одговара на поставените прашања поврзани со рецепти, користејќи `LangChain`, вектор база `FAISS`, модел `Mistral-7b-OpenOrca`.
- `contact()`: Рендерира форма за контакт, со што им се овозможува на корисниците да стапат во контакт со администраторите на `RecipeAI`.

6. URLs во апликацијата

- Во Django, датотеката `urls.py` служи како централна точка за мапирање на овие URL-ови кон соодветни функции за преглед, пружајќи јасен и организиран начин за ракување со барањата на корисниците и прикажување на соодветни одговори.
- Датотеката `urls.py` во оваа Django апликација дефинира низа од шаблони на URL-ови со цел навигација на корисниците и интеракција со различните компоненти на веб-сајтот.
- Почнувајќи со `admin` URL-от, корисниците со административни привилегии можат да пристапат до Django `admin`-интерфејсот за ефикасно управување

со содржината на сајтот. Во рамките на апликацијата имаме рути за сè што е потребно, односно покрај '/admin', постојат и ' ' → за почетната страница, '/login', '/register', '/logout', 'recipes', 'recipes/<int:recipe_id>/' → за прикажување детали за избран рецепт, 'add-recipe/', 'recipe/edit/<int:recipe_id>', 'for-you/', 'contact/', 'contact-form'. Исто така и 'delete-recipe/<int:recipe_id>/' каде што се предава ID на рецептот кој сакаме да го избришеме, оваа вредност ќе биде проследена на соодветната функција за преглед delete_recipe() како параметар, со што овозможува бришење на соодветниот рецепт во базата на податоци.

7. Templates во апликацијата

- Најпрвин го создадовме base.html темплејтот, од кој наследуваат сите останати. Овој темплејт претставува основа и ја вклучува главната навигација, односно токму во него се наоѓа navbar-от со линкови кон почетната страница, рецепти, персонализираните предлози, за нас и контакт. Вака дизајнираната навигација им овозможува на корисниците полесно да го најдат тоа што им е потребно. Што се однесува до стиловите, base.html ги дефинира општите стилови кои се користат на сите страници во апликацијата, односно ова ги вклучува стиловите за навигација, текстот, бојата на фонтовите и слично. Овој шаблон служи како почетна точка за секоја страница во апликацијата RecipeAI, обезбедувајќи конзистентно и атрактивно корисничко искуство преку целиот сајт.
- Index темплејтот (index.html) служи како почетна страница на нашата веб апликација. Првин е претставена привлечна слика која претставува тон за кулинарското патешествие кое ги очекува корисниците додека ја истражуваат веб-страницата. Секцијата "About Us" која се наоѓа во овој темплејт претставува краток преглед за RecipeAI, претставен преку слика и текст. Значајни карактеристики вклучуваат респонзивни дизајн елементи кои гарантираат оптимално гледање на различни уреди, зголемувајќи пристапност и ангажираност на корисниците. Исто така, на почетната страница се прикажува и одбрана селекција од популарни рецепти, што ги привлекува корисниците да ги истражат различните кулинарски понуди на платформата. Секој рецепт е дополнет со краток опис и слика, и при клик ќе одведе на детали за селектираниот рецепт.
- Шаблонот за најава (login.html) обезбедува структуриран интерфејс за автентикација на корисниците. Со чист и интуитивен дизајн, шаблонот го олеснува процесот на најава со тоа што ги поттикнува корисниците да ги внесат своите податоци во структурирана форма. Механизмите за валидација на форми вградени во Django помагаат во одржувањето на целосен интегритет на податоците, обезбедувајќи гладко и безбедно искуство на најава на корисниците.

- Шаблонот за најава (login.html) обезбедува структуриран интерфејс за автентикација на корисниците. Со чист и интуитивен дизајн, шаблонот олеснува процесот на најава со тоа што корисниците ги поттикнува да внесат своите податоци во структурирана форма. Механизмите за валидација на форми вградени во Django помагаат во одржувањето на целосната интегритет на податоците, обезбедувајќи гладоко и безбедно искуство на најава за корисниците.
- За новите корисници кои се расположени да се приклучат на кулинарската заедница, шаблонот за регистрација (registration.html) обезбедува едноставен начин за креирање на профил. Со јасно означени полиња за внесување на податоци како корисничко име, е-пошта и лозинка, шаблонот го поедноставува процесот на регистрација, придржувајќи се до најдобрите практики во дизајнот на кориснички интерфејс. Можностите за обработка на форми во Django ја олеснуваат валидацијата на податоците и давањето на повратни информации за грешки, остварувајќи регистрација без проблеми.
- Во шаблонот за рецепти (recipes.html), на корисниците им е претставен опширен каталог на вкусни рецепти, елегантно организирани во визуелно привлечен grid layout. Секоја картичка за рецепт е претставена со слика, име и краток опис на рецептот, како и копче 'See more', кое доколку се кликне ќе ги прикаже деталите за тој рецепт. Доколку е најавен корисникот, во овој темплејт се прикажува и копчето за додавање на нов рецепт.
- Шаблонот details.html овозможува детален преглед за избраниот рецепт. Ги прикажува сите податоци кои се дадени на рецептот при негово креирање, како и коментарите и оценките кои корисниците ги имаат оставено за рецептот. Покрај ова, доколку рецептот е креиран од корисникот што моментално го разгледува, има опција за уредување и бришење на рецептот.
- На крај, contact.html служи за остварување на комуникација помеѓу корисникот и администраторите. Преку контакт формата корисниците можат да ги пренесат своите прашања, предлози или забелешки до администраторите на веб-сајтот. Искористувајќи JavaScript за валидација на формата, шаблонот гарантира точност на податоците пред испраќање, зголемувајќи ја ефикасноста и надежноста на каналите за комуникација.

8. Forms во апликацијата

- Формите во нашата апликација имаат клучна улога во олеснувањето на интеракциите на корисниците. CustomAuthenticationForm го олеснува најавувањето со усовршување на изгледот на формата за најавување, преку Bootstrap стајлинг и прилагодување на label color. За регистрација на нов корисник, CustomUserCreationForm ја проширува нативната форма за регистрација на Django со плус опција за избор на role на корисникот.

- Кога станува збор за управување со рецепти, AddRecipeForm и EditRecipeForm го олеснуваат додавањето и уредувањето на рецепти, преку презентирање на структурирани интерфејси со Bootstrap стајлинг.
- На крај, ReviewForm им овозможува на корисниците да дадат свое мислење за рецептот, со полиња за оставање коментар и оценка покрај механизмите за валидација на точност на податоците.
- Заедно, овие форми ги усогласуваат интеракциите на корисниците, поттикнувајќи интуитивно и беспрекорно искуство во нашата апликација.