Autor: Mateusz Nowak

Uruchomienie wszystkich testów: należy zakomentować w pliku docker-compose.yml sekcję io-przepisy. Na komputerze musi być zainstalowany Python 3.7 wraz z wszystkimi modułami wymienionymi w pliku requirements.txt. Należy następnie uruchomić bazę danych przez polecenie docker-compose up w głównym katalogu projektu, a następnie wywołać:

python manage.py test

**1. Testy jednostkowe:**

Testy modeli:

Sprawdzono czy pola Recipe.title i Measurment.unit są unikalne, aby zagwarantować unikalność przepisu i jednostki miary. Ponadto przetestowane te modele, które miały więcej metod niż gettery i settery. W tym wypadku model Recipe posiadał metodę :

„get\_filtered\_recipes(ingredients, ingredients\_not, min\_time, max\_time ,difficulty\_level)”

Przeprowadzono testy sprawdzające filtrowanie:

* bez wymagań
* ze składnikami pożądanymi
* po składnikach niepożądanych
* po czasie przygotowania
* po poziomie trudności

Testy kontrolerów:

* home

Sprawdzono, czy znane składniki są przekazywane do widoku.

* recipe\_detail

Sprawdzono, czy przepis zostaje przekazany do widoku

* get\_difficulty\_level

Sprawdzono czy funkcja zwraca odpowiednią wartość na podstawie nazw poziomów występujących w aplikacji i losowych ciągów znaków.

* recipe\_list

Sprawdzono rozpoznawanie przez metodę przekazanych składników.

Sprawdzono zachowanie w przypadku nie znalezienia składnika.

* update\_mark

Sprawdzono aktualizację oceny dla wartości z zakresu akceptowanego i z poza niego.

Sprawdzono dzielenie przez 0.

**2. Testy integracyjne:**

Testowano integrację kontrolerów z modelami.

Sprawdzono:

* dodanie oceny do przepisu
* filtrowanie przepisów
* wyszukiwanie przepisu

**3. Cel testów**

Testy nie pokrywają wszystkich możliwych sytuacji użycia tych metod, ale pozwoliły znaleźć kilka błędów. Głównym ich celem była minimalizacja sytuacji gdy zmiana metody zepsuje nam całą aplikację.

**4. Znalezione błędy**

|  |  |  |
| --- | --- | --- |
| **GDZIE** | **CO** | **WAŻNOŚĆ** |
| get\_filtered\_recipes | Zwracała wszystkie przepisy których nazwa składników chociaż w części zawierała pożądany(lub niepożądany) składnik | 1 |
| recipe\_list | Błąd, jeżeli żądanie nie zawiera w tablicy GET wszystkich 6 składników | 3 |
| recipe\_list | Błąd, jeżeli nie ma w Tablicy GET „min-time” lub „max-time” | 3 |
| get\_difficulty\_level | Błąd, jeżeli w tablicy GET nie ma „difficulty-level” | 3 |
| update\_mark | Metoda dopuszczała dzielnie przez 0 i nie obsługiwała wyjątku | 2 |
| update\_mark | Błąd, jeżeli żądanie nie zawiera w tablicy GET „recipe-id” lub „mark” | 3 |

Poziomy ważności:

1. Krytyczne – zagrażają bezpośrednio działaniu aplikacji i bezpieczeństwu. Wymagają natychmiastowej poprawy.
2. Ważne – aplikacja pomimo błędów zazwyczaj działa. Naprawa obowiązkowa.
3. Średnio-ważne – błędy nie są możliwe do wywołania w obecnej formie aplikacji z poziomu tejże lub przypadki ich wystąpienia są mało realne. Naprawa zalecana.