Національний технічний університет України

Київський політехнічний інститут ім. Ігоря Сікорського

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**ЗВІТ**

**ПРО ПРОХОДЖЕННЯ ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ**

Студента(ки) \_\_\_4\_\_ курсу \_\_ІТ-83\_\_\_ групи

Спеціальності 121 «Інженерія програмного забезпечення»

\_\_Бєкішєв \_Ілля \_Євгенійович \_

(прізвище, ім’я, по батькові студента)

Термін практики з « 2 \_» 05\_ 2022\_ р. по «\_29\_\_»05\_2022\_ р.

База практики: \_\_ кафедра ІПІ, КПІ ім. Ігоря Сікорського \_ \_ \_

(назва підприємства)

Керівник від підприємства

\_\_доцент\_ \_

(науковий ступінь, вчене звання, посада)

\_ Ліхоузова Тетяна Анатоліївна \_ \_

(прізвище, ім’я, по батькові, підпис)

Керівник практики від кафедри ІПІ

\_\_старший викладач\_ \_

(науковий ступінь, вчене звання, посада,)

\_\_ Марченко Олена Іванівна \_ \_

(прізвище, ім’я, по батькові, підпис)

**Київ-2022р.**

**Пояснювальна записка  
до дипломного проекту**

на тему: Додаток для складання збалансованого меню

КПІ.ІТ-7304.045440.02.81

Київ – 2022

зміст

1.1 Опис та аналіз предметної області 5

1.2 Аналіз успішних IT-проектів 5

1.3 Аналіз вимог до програмного забезпечення 10

1.4 Постановка задачі 12

Висновки до розділу 12

2.1 Моделювання та аналіз програмного забезпечення 13

2.2 Архітектура програмного забезпечення 18

2.3 Структура даних та ресурсів програми 19

2.4 Аналіз безпеки даних 50

Висновки до розділу 51

3.1 Аналіз якості ПЗ 52

3.2 Опис процесів тестування 53

3.3 Опис контрольного прикладу 54

Висновки до розділу 61

4.1 Розгортання програмного забезпечення 62

4.2 Робота з програмним забезпеченням 63

Висновки до розділу 68

Перелік умовних позначень

|  |  |  |
| --- | --- | --- |
| IDE | – | Integrated Development Environment – інтегроване середовище розробки. |
| API | – | Application programming interface, прикладний програмний Інтерфейс |
| SDK | – | Software development kit |
| IT | – | Інформаційні технології |
| ER | – | Entity-Relation diagram |
| OC | – | Операційна система. |
| СУБД | – | Системи управління базами даних. |
| БД | – | База даних. |

Вступ

Збалансоване харчування – запорука міцного здоров’я, хорошого самопочуття та довголіття. Тому кожна людина рано чи пізно починає звертати увагу, чи корисну їжу вживає. Хтось іде до дієтолога, хтось читає кулінарні книги, але найбільше людей шукають відповіді в інтернеті. Саме їм на допомогу стають різні застосунки, призначені для складання меню.

Враховуючи те що складання меню збалансованого харчування за допомогою різних застосунків не дійшло до стану коли можливо при натисканні декількох кнопок отримати меню з списком блюд на ту суму і ту кількість днів яку хотів би користувач, а також побачити в яких саме магазинах зараз можливо придбати ці продукти. Я вирішив створити власний веб-додаток який би був спроможний на це.

Таким чином можливо створити людям більше вільного часу на повсякденні діла та понизити поріг входу до здорового харчування завдяки алгоритмам які самі розрахують меню збалансованого харчування.

Далі завдяки такому алгоритму можливо буде наприклад створити новий тип роботи для людей які вміють готувати їжу за рецептами. Саме передавати їм меню для конкретної людини по її заказу, що могло б ще більше звільнити час людини та надати нові робочі місця.

Тому за мету, об’єкт дослідження та предмет дослідження були обрані наступні теми.

Мета: надати можливість пересічному користувачу самостійно складати меню для збалансованого харчування.

Об’єкт дослідження: програмне забезпечення для створення меню.

Предмет дослідження: методи та алгоритми формування меню.

Завданням роботи є створення додатку для формування меню збалансованого харчування.

# АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Опис та аналіз предметної області

Для створення меню збалансованого харчування людині потрібно витратити багато часу для ознайомлення з усіма існуючими методами вирішення проблеми забезпечення себе та своєї сім’ї належним харчуванням та за таку ціну, яка була би підходящою саме для бюджету цієї конкретної людини. Таким чином, спочатку складене людиною меню може не підходити йому по бюджету. Для кожної людини потрібно підбирати меню харчування відповідно до потреб його організму, а не підготовлений бажаючий правильно харчуватися не знає про багато нюансів таких як група по навантаженню, до якої він відноситься і те, що регіон його проживання теж впливає на те, як йому слід харчуватися. Час, витрачений на похід до фахівця, не завжди може бути оптимально витрачений, адже лікар може видати вам меню, яке буде не підходити під ваш гаманець і набриднути вам через певний час. Саме тому було б гарно, якщо був би засіб створення меню збалансованого харчування, який би міг задовольнити усі ці потреби.

## Аналіз успішних IT-проектів

На даний момент програми для створення меню збалансованого харчування ще не дійшли до такого стану, щоб в одному застосунку були представлені можливості для отримання меню за ту ціну, яку користувач хотів би витратити на нього, та швидкого знаходження місця, де можливо придбати продукти для приготування блюд з меню.

Існують декілька варіантів програм для виконання даної цілі. Перша це програми, які надають можливість самостійно обирати, що ви хотіли би з’їсти, та додати ці блюда до меню та самостійно впевнитись в тому, що результат буде відповідати нормам збалансованого харчування. Також тільки в даному виді програм мною було виявлено функцію зберігання усього списку їжі, яку користувач з’їв. Другий варіант – програми, які дозволяють автоматично створювати меню за допомогою натискання кількох кнопок та введення інформації про стан свого тіла. Але я не знайшов жодного програмного рішення, яке дозволяє автоматично створити меню з вказанням бюджету на певний період часу так, щоб було ясно де саме можливо купити ці продукти.

Критерії, за якими будемо порівнювати існуючі рішення:

* можливість використання додатку на різних пристроях;
* можливість ведення книги рецептів;
* можливість оцінки калорійності окремого продукту/блюда в цілому за його складом;
* можливість підбору блюд з врахуванням обмежень по складовим/калорійності/вартості/кількості порцій;
* можливість перегляду цін на необхідні продукти у найближчих точках продажу;
* можливість замовлення/придбання переліку необхідних продуктів у найближчих точках продажу.

### Аналіз технічних рішень

Технічним рішенням для такого програмного забезпечення може бути веб-додаток або мобільний додаток. У кожного варіанта є свої переваги та недоліки. Веб-додаток кращий та зручніший тому що його можливо запускати на будь якому пристрої в якого є браузер.

### Аналіз програмних продуктів

**Сайт «price.ua»** [3]. Вигляд основної сторінки зображено на рисунку 1.1.

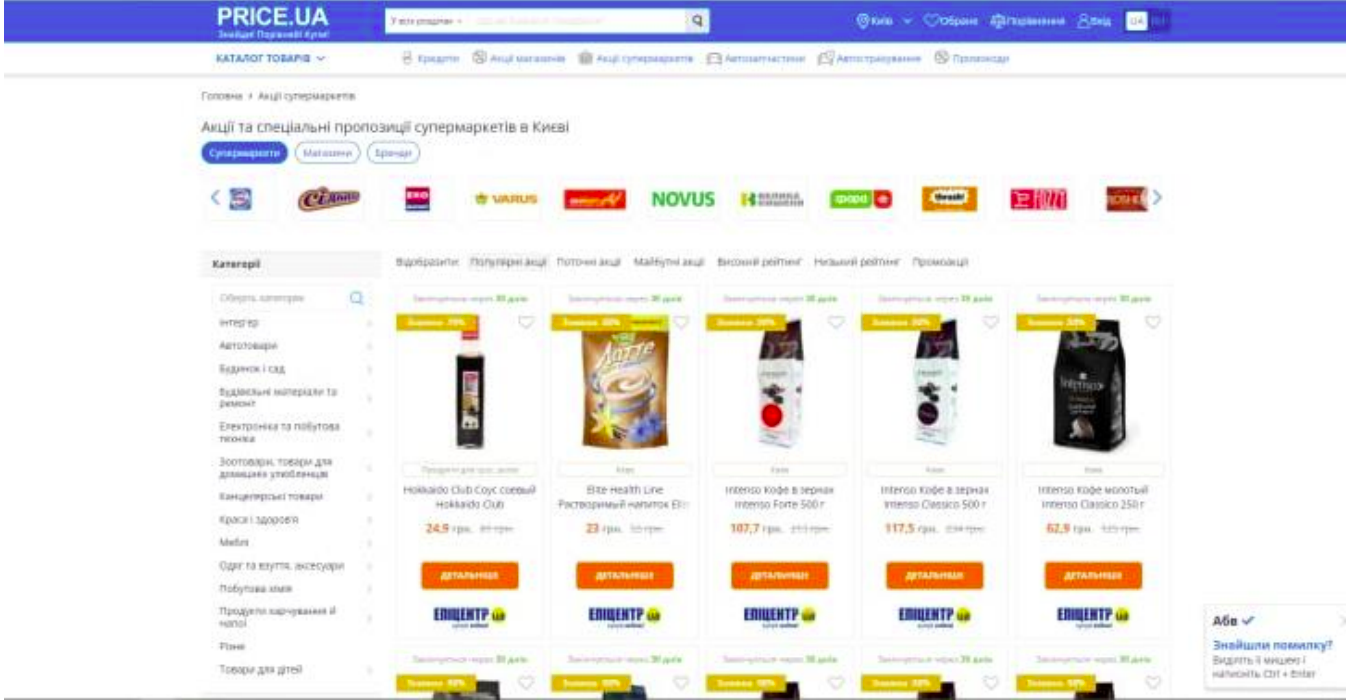


Рисунок 1.1 – Головна сторінка сайту «price.ua»

Ліва частина меню містить зручний вибір категорії товарів, де відображено багато різних популярних категорій продуктів. На сайті відображені поточні мінімальні ціни на продукти, проте функціонал не дозволяє створити кошик необхідних товарів і оцінити їх сумарну вартість. Отже, цей сайт добре підходить для разового використання на одиночні покупки товарів. Проте людині, що турбується за поживність речовин своєї їжі, буде складно підібрати збалансоване харчування за оптимальною ціною.

**Додаток для мобільних телефонів «Kalorické Tabulky»** [4]. Вигляд основних сторінок наведено на рисунку 1.2.

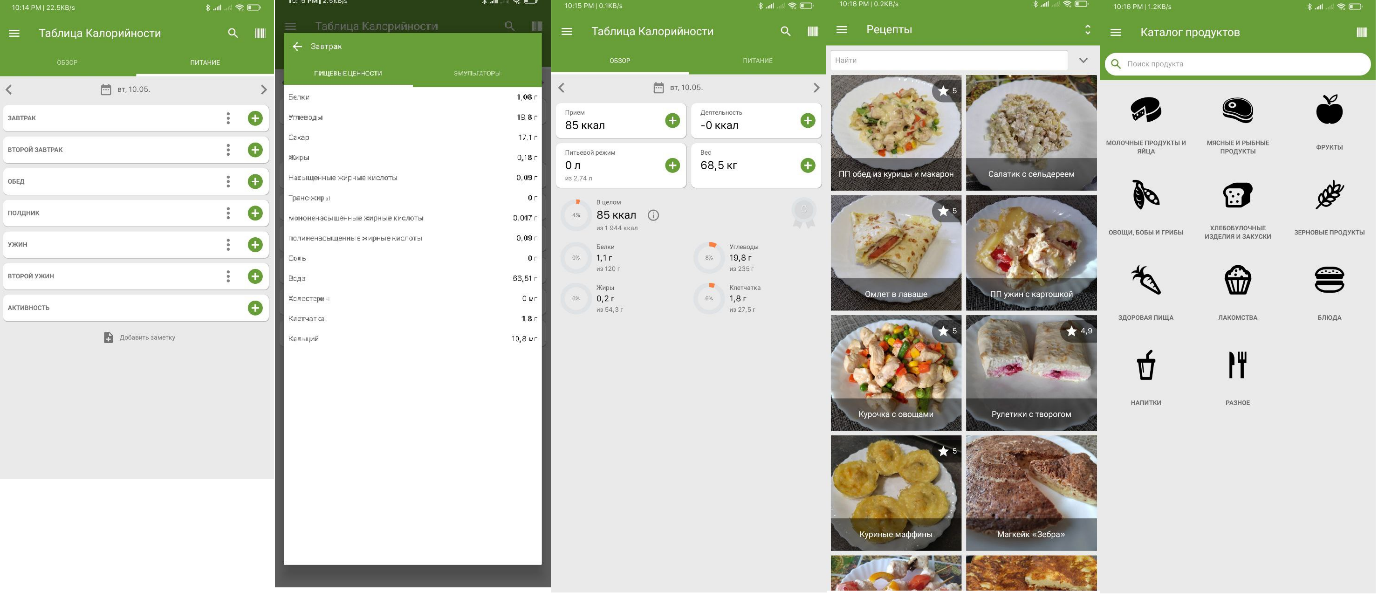


Рисунок 1.2 – Основні сторінки «Kalorické Tabulky»

В додатку присутня можливість перегляду калорійності продукту, можливість додавати продукти в меню на день, а також перегляд існуючих у базі даних блюд, які теж можливо додати до меню. Якщо користувач додав продукт чи блюдо до меню, то в його календарі споживання їжі з’явиться додана страва та збільшиться кількість споживаних калорій за цей день. Отже така програма буде гарною для складання меню харчування власноруч та надає можливість стежити за споживаними калоріями.

**Мобільний додаток «Mary’s Recipes»** [5]. Вигляд основних сторінок наведено на рисунку 1.3.

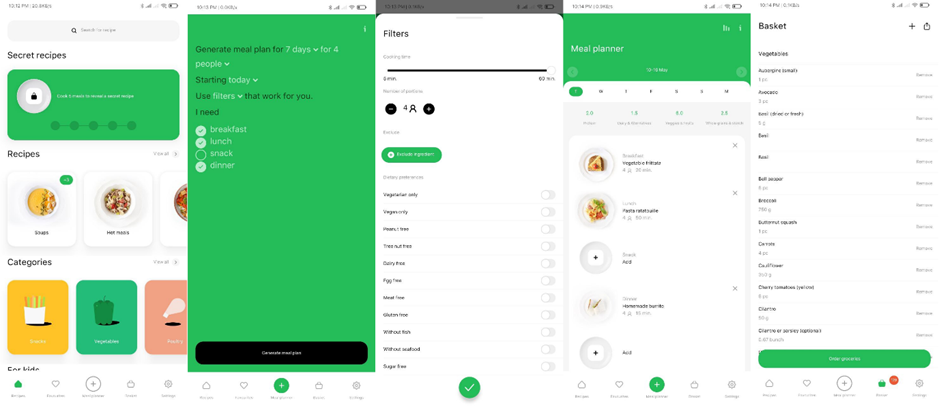


Рисунок 1.3 – Основні сторінки «Mary’s Recipes»

Дана програма може скласти меню на кілька днів для кількох людей з урахуванням декількох видів прийому їжі з можливістю прибрати небажані продукти. Також можливо додавати продукти зі складеного меню до списку продуктів. Але цей додаток працює по підписці, яка надає доступ до деяких закритих в безкоштовній версії блюд та можливості відображати харчову цінність.

Таблиця 1.1 – Порівняння існуючих рішень

|  |  |  |  |
| --- | --- | --- | --- |
| Критерій | price.ua | Kalorické Tabulky | Mary’s Recipes |
| можливість використання додатку на різних пристроях | + | + | + |
| можливість ведення книги рецептів | - | + | - |
| можливість оцінки калорійності окремого продукту/блюда в цілому за його складом | - | + | +/- |
| можливість підбору блюд з врахуванням обмежень по: |  |  |  |
| * складовим | - | - | + |
| * калорійності | - | + | + |
| * вартості | + | - | - |
| * кількості порцій | + | - | - |
| можливість перегляду цін на необхідні продукти у найближчих точках продажу | + | - | - |
| можливість замовлення/придбання переліку необхідних продуктів у найближчих точках продажу | - | - | - |

Аналіз програмних продуктів виявив, що на даний момент не існує таких проектів, які б вдовольняли усім критеріям. Але цього можливо досягнути, якщо використовувати одразу декілька програм із різних ІТ-проектів.

## Аналіз вимог до програмного забезпечення

Спочатку опишемо бажані можливості програмного забезпечення, що має вирішувати проблему. В застосунку повинна бути можливість створювати магазини зі списком продуктів які в ньому присутні та цінами на них. У кожного продукту повинна бути інформація про кількість білків, жирів, вуглеводів, вітамінів та мінералів присутніх в ньому. Особливо важливим є присутність каталогу рецептів з можливістю додавання власних рецептів з вказанням продуктів з бази даних та кухонних приладь потрібних для цього блюда, також повинна бути можливість швидко знайти де купити потрібні продукти. Також невід’ємною частиною є реєстрація в додатку для заповнення інформації про стан тіла користувача. Для подальшого покращення програмного застосунку потрібно реалізувати можливість додавання блюд та продуктів до списку обраного, щоб у майбутньому можливо було персоналізувати меню збалансованого харчування. Для того щоб користувач в подальшому зміг створювати меню тільки з тими кухонними приладдями які в нього є потрібно створити можливість додавати кухонне приладдя до бази даних. Самою важливою частиною є створення алгоритму який би зміг скласти меню збалансованого харчування на потрібну суму та кількість днів і щоб це робилося за мінімальну кількість рухів.

Оскільки в цьому переліку «побажань» досить багато, є сенс виділити серед них ті, що мають бути обов’язково реалізовані в застосунку.

### Розроблення функціональних вимог

Програмний продукт має забезпечити виконання наступних функцій:

* Реєстрація у додатку.
* Ведення каталогу продуктів та рецептів:
  + додавання продуктів з інформацією про їх поживну цінність;
  + додавання рецептів з існуючих у БД продуктів;
* Створення меню збалансованого харчування з врахуванням обмежень по складовим, калорійності, вартості, кількості порцій:
  + списки обраних блюд;
  + списки обраних продуктів.
* Ведення каталогу магазинів
  + додавання магазинів.

Функціональні вимоги, що є додатковими:

* Інформація про поточний стан здоров’я користувача (перелік обмежень на продукти).
* Список присутніх у користувача кухонних приладь.
* Каталог з цінами на продукти в окремих магазинах.

### Розроблення нефункціональних вимог

При використанні даної програми у користувача повинен зменшитись час, який він витрачає на складання меню збалансованого харчування, таким чином в користувача з’являється більше вільного часу. Також повинні мінімізуватися затрати на харчування. Зменшитись час який витрачається на знаходження потрібних для меню продуктів в магазинах. Повинне буде покращитись стан здоров’я користувача при постійному споживанні блюд із наданих програмою меню збалансованого харчування. Отже, нефункціональні вимоги полягають в наступному:

* Мова додатку - українська.
* Підтримується використання найбільш популярних браузерів.
* Інтуїтивно зрозумілий інтерфейс користувача.

## Постановка задачі

Метою розробки є надати можливість пересічному користувачу самостійно складати меню для збалансованого харчування. Для цього потрібно вирішити наступні задачі:

* створити каталоги продуктів, рецептів, магазинів;
* спроектувати та розробити модуль формування меню;
* спроектувати та розробити модуль оцінки кількості/вартості продуктів для заданого меню.

## Висновки до розділу

При написанні цього розділу було виділено проблеми, що необхідно вирішити, а також визначені об’єкт та предмет дослідження. Проведено загальний огляд об’єкта дослідження. Проаналізовані існуючі рішення, їх переваги та недоліки. Визначено функціональні та нефункціональні вимоги до програмного забезпечення, що розробляється.

# МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Моделювання та аналіз програмного забезпечення

Першим кроком при проектуванні програмного забезпечення є розуміння, як саме воно буде використовуватись. В даному випадку ініціатором всіх дій, що виконуватиме ПЗ, є людина-користувач. Тому почнемо з опису взаємодії користувача та ПЗ, для чого скористаємось UML-діаграмою.

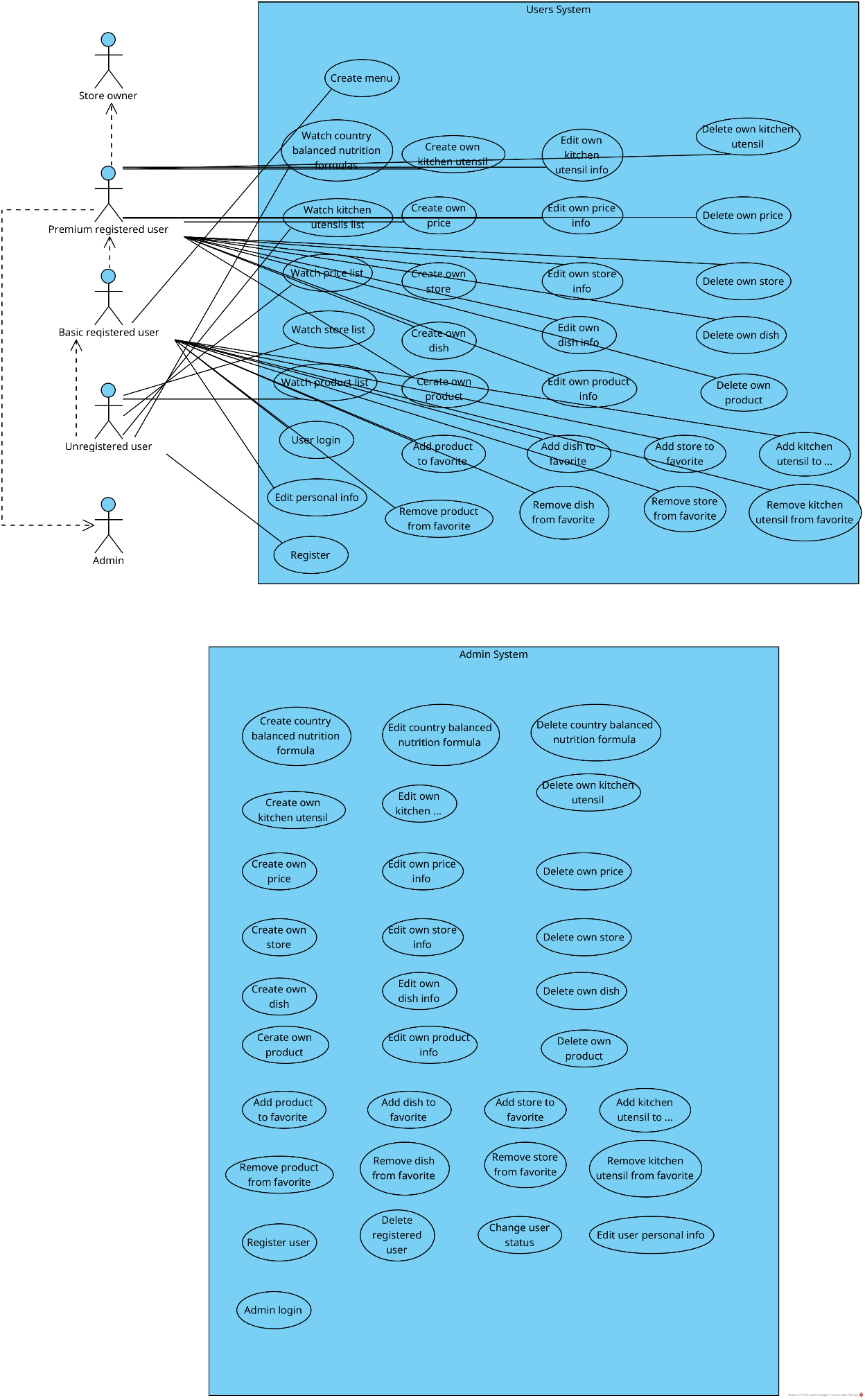


Рисунок 2.1 – Схема структурна варіантів використання

Для наочності опису основних компонентів використовуємо діаграму компонентів. Вона дозволяє побачити взаємодію між розробленими шарами веб-додатку. Діаграма компонентів відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного забезпечення може бути представлено як компоненту. Деякі компоненти існують під час компіляції, деякі — під час компонування, а деякі під час роботи програми.

Діаграма компонентів для розробленого веб-застосунку зображена на рисунку 2.2

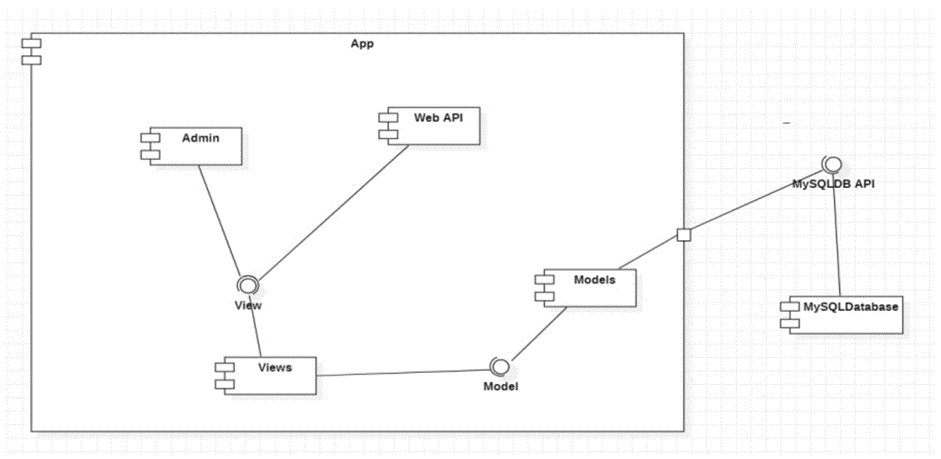


Рисунок 2.2 – Діаграма компонентів

### Модуль створення меню

Модуль створення меню складається з декількох частин, а саме:

* підрахунок корисних речовин в блюді окремо та у меню на один день;
* перевірка на те чи меню задовольняє стандартам харчування в регіоні проживання користувача;
* підрахунок вартості меню.

Кожен з цих модулів потрібен для того щоб розрахувати меню збалансованого харчування на ту суму грошей яка є у користувача та на ту кількість днів яка йому потрібна.

Блок-схему роботи алгоритму наведено на рисунку 2.3

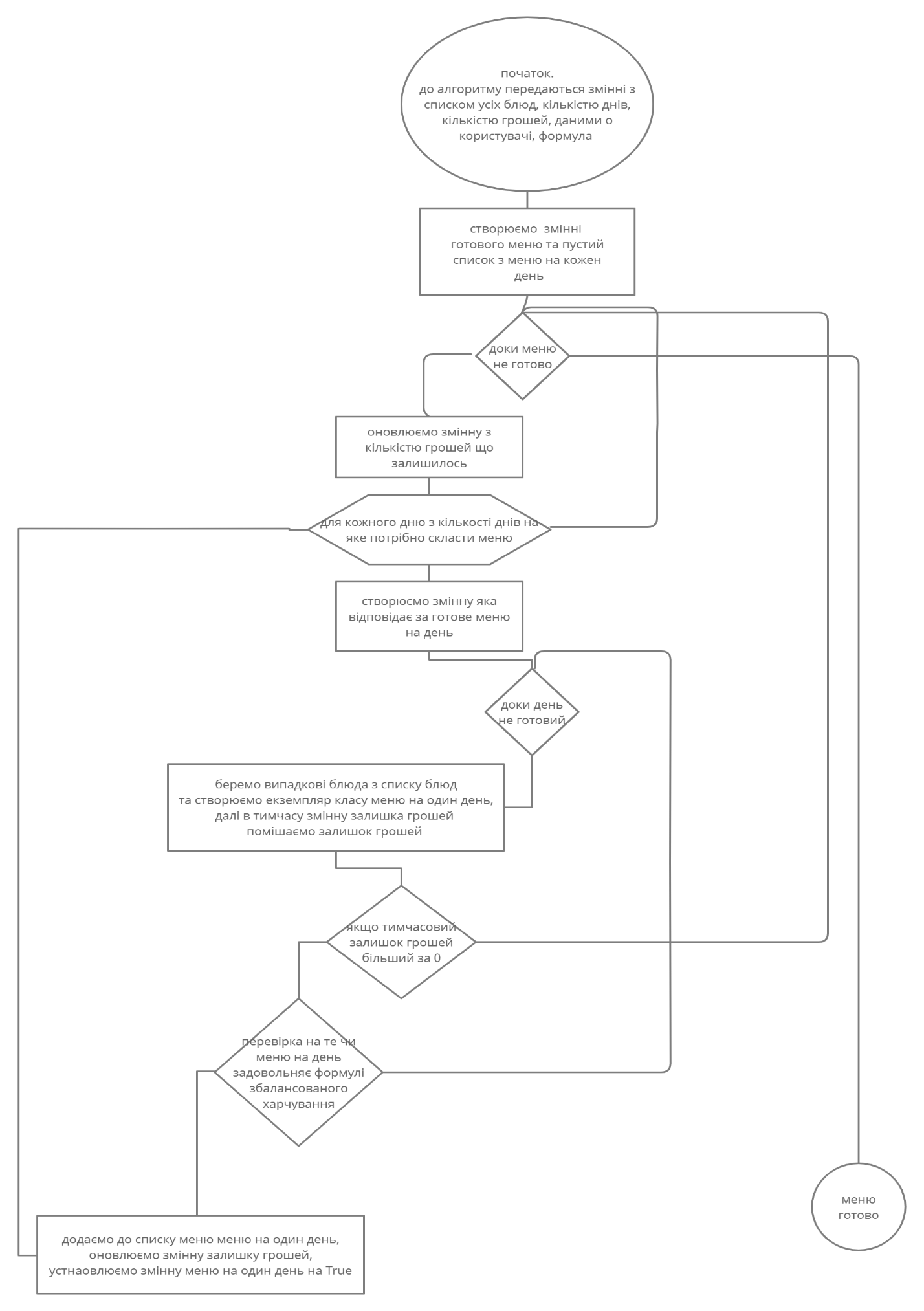


Рисунок 2.3 – блок-схема роботи алгоритму

## Архітектура програмного забезпечення

Зазвичай фреймворки веб-додатків використовують архітектуру Model-View-Controller (MVC). MVC популярний, оскільки він ізолює логіку програми від рівня інтерфейсу користувача, і різні частини програми можна розробляти окремо. Проте за правильної реалізації Model-View-Template (MVT) розробник може не турбуватися про те, як дані переміщуються між моделлю і представленням, що значно прискорює процес розробки ПЗ. Саме тому для реалізації проекту будо вибрано архітектуру MVT.

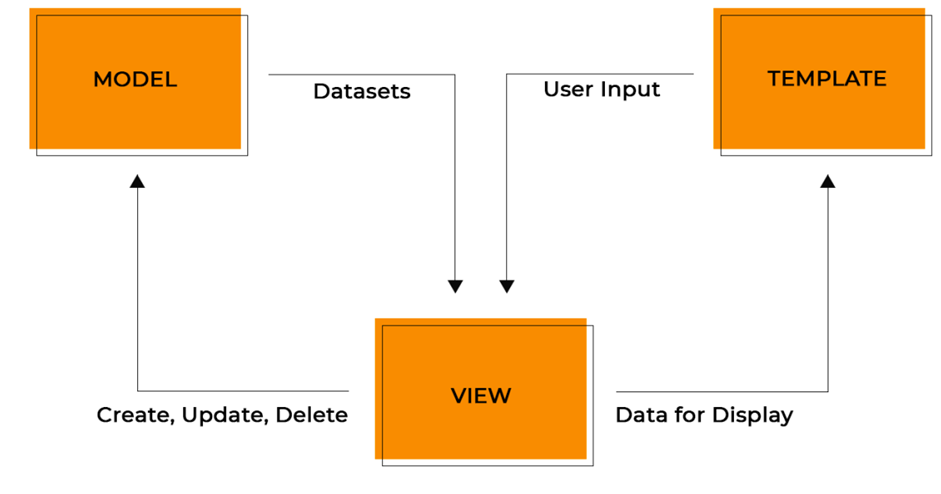


Рисунок 2.4 – Архітектура MVT

MVT архітектура складається з трьох наступних частин.

Модель: модель буде діяти як інтерфейс ваших даних. Він відповідає за збереження даних. Це логічна структура даних, що лежить в основі всієї програми, і представлена базою даних.

Представлення: це інтерфейс користувача — те, що ви бачите у своєму браузері, коли відкриваєте веб-сайт. Він представлений файлами HTML/CSS/Javascript.

Шаблон: шаблон складається зі статичних частин потрібного виводу HTML, а також певного спеціального синтаксису, що описує, як буде вставлений динамічний вміст.

## Структура даних та ресурсів програми

Додаток містить три основних модулі:

* Core
* Templates
* Catalog



Рисунок 2.5 – Структура проекту

Таблиця 2.1 – Таблиця опису класів

|  |  |
| --- | --- |
| Клас | Опис |
| Product\_database/manage | Інтерфейс для django-admin |
| Product\_database/catalog/admin | Опис та додавання елементів до панелі адміністратора |
| Product\_database/catalog/apps | Конфігурація додатку |
| Product\_database/catalog/bnfmc | Підрахунок меню збалансованого харчування на певний період та на певну суму грошей |
| Product\_database/catalog/cdn | Підрахунок даних о харчуванні |
| Product\_database/catalog/choices | Файл з конфігураціями вибіркових даних |
| Product\_database/catalog/citmmtf | Перевірка на те чи задовільняе меню параметрам для окремого користувача |
| Product\_database/catalog/cp | Підрахунок вартості меню |
| Product\_database/catalog/create\_regions\_models | Створення пустих таблиць для кожного регіону з вказанням на потрібний для споживання набор корисних елементів |
| Product\_database/catalog/data\_dict | Дані о регіонах та їх знаходженні |
| Product\_database/catalog/forms | Створення форм для заповнення даних на сайті |
| Product\_database/catalog/human\_attributes\_template | Створення макету атрибутів корисного харчування |
| Product\_database/catalog/models | Створення моделей в БД |
| Product\_database/catalog/regions | Отримання даних про назви регіонів з фалу Product\_database/catalog/data\_dict |
| Product\_database/catalog/tests | Тести |
| Product\_database/catalog/urls | Створення патернів посилань на сторінки на сайті |
| Product\_database/catalog/vievs | Створення методів відображення та валідації даних на сторінках |
| Product\_database/catalog/tests/test\_forms | Тести для форм відправки даних |
| Product\_database/catalog/tests/test\_models | Тести моделей з БД |
| Product\_database/catalog/tests/test\_views | Тести методів відображення та валідації даних |
| Product\_database/product\_dabase/asgi | Налаштування клієнт-серверного протоколу взаємодії веб-сервера та додатку |
| Product\_database/product\_dabase/settings | Налаштування проекту |
| Product\_database/product\_dabase/urls | Створення паттернів посилань на сторінки на сайті |
| Product\_database/product\_dabase/wsgi | Налаштування стандарту взаємодії Python-додатку з веб-сервером |

Таблиця 2.2 – Таблиця опису методів класів

|  |  |  |
| --- | --- | --- |
| Клас | Метод | Опис |
| Product\_database/manage | main | Налаштування оточення для Django |
| Product\_database/catalog/admin | HumanAttributesAdmin | Реєстрація моделі HumanAttributes у адміністративній частині додатку |
| Product\_database/catalog/admin | HumanAttributesInline | Додавання можливості зручного користування у адміністративній частині додатку до моделі HumanAttributes |
| Product\_database/catalog/admin | BalancedNutritionFormulaAdmin | Реєстрація моделі BalancedNutritionFormula у адміністративній частині додатку |
| Product\_database/catalog/admin | ProfileAdmin | Реєстрація моделі Profile у адміністративній частині додатку |
| Product\_database/catalog/admin | PriceInline | Додавання можливості зручного користування у адміністративній частині додатку до моделі Price |
| Product\_database/catalog/admin | FavoriteInline | Додавання можливості зручного користування у адміністративній частині додатку до моделі Favorite |
| Product\_database/catalog/admin | PriceAdmin | Реєстрація моделі Price у адміністративній частині додатку |
| Product\_database/catalog/admin | ProductAdmin | Реєстрація моделі Product у адміністративній частині додатку |
| Product\_database/catalog/admin | StoreAdmin | Реєстрація моделі Store у адміністративній частині додатку |
| Product\_database/catalog/admin | ProductAmountAdmin | Реєстрація моделі ProductAmount у адміністративній частині додатку |
| Product\_database/catalog/admin | KitchenUtensilAdmin | Реєстрація моделі KitchenUtensil у адміністративній частині додатку |
| Product\_database/catalog/admin | DishAdmin | Реєстрація моделі Dish у адміністративній частині додатку |
| Product\_database/catalog/apps | CatalogConfig | Конфігурація додатку |
| Product\_database/catalog/bnfmc | balanced\_nutrition\_formula\_menu\_calculation | Підрахунок меню збалансованого харчування |
| Product\_database/catalog/cdn | create\_empty\_nutrition\_data | Створення пустої моделі компонентів їжі для подальшого використання |
| Product\_database/catalog/cdn | calculate\_dish\_nutrition | Підрахунок усіх поживних компонентів блюда |
| Product\_database/catalog/cdn | calculate\_nutrition\_for\_menu\_for\_one\_day | Підрахунок усіх поживних компонентів меню на один день |
| Product\_database/catalog/choices | UNITS | Міри ваги продукту |
| Product\_database/catalog/choices | SEXES | Стать людини |
| Product\_database/catalog/choices | PAGS | Групи за фізичним навантаженям |
| Product\_database/catalog/choices | BIRTH\_YEAR\_CHOICES | Дозволені роки народження |
| Product\_database/catalog/choices | KID\_YEAR\_LIST | Категорії віку дитини |
| Product\_database/catalog/choices | YEAR\_LIST | Інші категорії віку |
| Product\_database/catalog/choices | ALL\_YEARS | Усі категорії віку |
| Product\_database/catalog/choices | WEIGHT | Вагові категорії дорослої людини |
| Product\_database/catalog/citmmtf | check\_if\_the\_menu\_matches\_the\_formula | Перевірка чи співпадає меню з запропонованими стандартами харчування в регіоні проживання користувача |
| Product\_database/catalog/cp | calculate\_price | Підрахунок вартості меню в регіоні користувача |
| Product\_database/catalog/create\_regions\_models |  | Створення пустих таблиць для кожного регіону з вказанням на потрібний для споживання набор корисних елементів |
| Product\_database/catalog/forms | DataForCalculatingTheFormulaForm | Форма для збору даних для підрахунку меню збалансованого харчування |
| Product\_database/catalog/forms | HumanAttributesForm | Форма для збору даних о атрибутах для харчування |
| Product\_database/catalog/forms | UserEditForm | Форма для редагування даних користувача |
| Product\_database/catalog/forms | ProfileEditForm | Форма для редагування даних користувача |
| Product\_database/catalog/forms | UserRegistrationForm | Форма реєстрації користувача |
| Product\_database/catalog/forms | AddProductForm | Форма для додавання продукту |
| Product\_database/catalog/forms | AddStoreForm | Форма для додавання магазину |
| Product\_database/catalog/forms | AddPriceForm | Форма для додавання ціни на продукт |
| Product\_database/catalog/forms | RenewProductForm | Форма для редагування продукту |
| Product\_database/catalog/forms | RenewStoreForm | Форма для редагування магазину |
| Product\_database/catalog/forms | RenewPriceForm | Форма для редагування ціни |
| Product\_database/catalog/forms | ProductAmountForm | Форма для додавання кількості продукту |
| Product\_database/catalog/forms | AddKitchenUtensilForm | Форма для додавання кухонних приладь |
| Product\_database/catalog/forms | RenewKitchenUtensilForm | Форма для редагування кухонних приладь |
| Product\_database/catalog/forms | DishForm | Форма для блюда |
| Product\_database/catalog/human\_attributes\_template | create\_templates | Створення макету атрибутів корисного харчування |
| Product\_database/catalog/models | NutritionData | Модель для прискорення роботи з даними про харчову цінність продукту |
| Product\_database/catalog/models | MenuForMultipleDays | Модель меню на декілька днів |
| Product\_database/catalog/models | MenuForOneDay | Модель меню на один день |
| Product\_database/catalog/models | BalancedNutritionFormula | Модель формули збалансованого харчування |
| Product\_database/catalog/models | HumanAttributes | Модель атрибутів для корисних речовин для людини певної категорії |
| Product\_database/catalog/models | Profile | Модель профілю користувача на сайті |
| Product\_database/catalog/models | Product | Модель продукту |
| Product\_database/catalog/models | Store | Модель магазину |
| Product\_database/catalog/models | Price | Модель ціни |
| Product\_database/catalog/models | ProductAmount | Модель кількості продукту |
| Product\_database/catalog/models | KitchenUtensil | Модель кухонних приладь |
| Product\_database/catalog/models | Dish | Модель блюда |
|  | \_\_str\_\_ | Функція перетворення вибраного поля в строковий формат |
|  | get\_absolute\_url | Функція надання посилання на сторінку з детальною інформацією о моделі в БД |
| Product\_database/catalog/regions | get\_regions | Отримання доступних регіонів |
| Product\_database/catalog/urls | urlpatterns | Список патернів формувань посилань на сторінки на сайті |
| Product\_database/catalog/vievs | edit\_profile | Передачі інформації на сторінку редагування профілю користувача |
| Product\_database/catalog/vievs | register | Передача даних на сторінку реєстрації користувача |
| Product\_database/catalog/vievs | index | Передача даних на головну сторінку |
| Product\_database/catalog/vievs | ProductListView | Передача даних на сторінку перегляду усіх продуктів |
| Product\_database/catalog/vievs | StoreListView | Передача даних на сторінку перегляду усіх магазинів |
| Product\_database/catalog/vievs | PriceListView | Передача даних на сторінку перегляду усіх цін |
| Product\_database/catalog/vievs | DishListView | Передача даних на сторінку перегляду усіх блюд |
| Product\_database/catalog/vievs | KitchenUtensilListView | Передача даних на сторінку перегляду усіх кухонних приладь |
| Product\_database/catalog/vievs | ProfileDetailView | Передача даних на сторінку детальної інформації профілю користувача |
| Product\_database/catalog/vievs | KitchenUtensilDetailView | Передача даних на сторінку детальної інформації кухонних приладь |
| Product\_database/catalog/vievs | DishDetailView | Передача даних на сторінку детальної інформації блюда |
| Product\_database/catalog/vievs | ProductDetailView | Передача даних на сторінку детальної інформації продукту |
| Product\_database/catalog/vievs | StoreDetailView | Передача даних на сторінку детальної інформації магазину |
| Product\_database/catalog/vievs | UsersFavoriteProductsListView | Передача даних на сторінку перегляду усіх продуктів які користувач помітив як обрані |
| Product\_database/catalog/vievs | UsersFavoriteDishesListView | Передача даних на сторінку перегляду усіх блюд які користувач помітив як обрані |
| Product\_database/catalog/vievs | UsersProductsListView | Передача даних на сторінку перегляду усіх продуктів які користувач додав до сайту |
| Product\_database/catalog/vievs | UsersDishesListView | Передача даних на сторінку перегляду усіх блюд які користувач додав до сайту |
| Product\_database/catalog/vievs | UsersKitchenUtensilsListView | Передача даних на сторінку перегляду усіх кухонних приладь які користувач додав до сайту |
| Product\_database/catalog/vievs | UsersStoresListView | Передача даних на сторінку перегляду усіх магазинів які користувач додав до сайту |
| Product\_database/catalog/vievs | UsersPricesListView | Передача даних на сторінку перегляду усіх цін які користувач додав до сайту |
| Product\_database/catalog/vievs | add\_product\_to\_favorite | Додавання продукту до списку обраних |
| Product\_database/catalog/vievs | delete\_product\_from\_favorite | Видалення продукту до списку обраних |
| Product\_database/catalog/vievs | add\_dish\_to\_favorite | Додавання блюда до списку обраних |
| Product\_database/catalog/vievs | delete\_dish\_from\_favorite | Видалення блюда з списку обраних |
| Product\_database/catalog/vievs | renew\_product | Перевірка даних при оновленні інформації о продукті |
| Product\_database/catalog/vievs | renew\_store | Перевірка даних при оновленні інформації о магазині |
| Product\_database/catalog/vievs | renew\_price | Перевірка даних при оновленні інформації о цині |
| Product\_database/catalog/vievs | create\_dish | Перевірка даних при додаванні блюда до БД |
| Product\_database/catalog/vievs | BalancedNutritionFormulaDetailView | Передача даних на сторінку детальної інформації формулах збалансованого харчування |
| Product\_database/catalog/vievs | BalancedNutritionFormulaListView | Передача даних на сторінку перегляду усіх формул збалансованого харчування |
| Product\_database/catalog/vievs | calculate\_formula | Передача даних для підрахунку меню збалансованого харчування |
| Product\_database/catalog/vievs | renew\_balanced\_nutrition\_formula | Перевірка даних при оновленні інформації о формулі збалансованого харчування |
| Product\_database/catalog/vievs | renew\_dish | Перевірка даних при оновленні інформації о блюді |
| Product\_database/catalog/vievs | DishDelete | Передача даних при видаленні блюда з БД |
| Product\_database/catalog/vievs | KitchenUtensilCreate | Передача даних при додаванні кухонних приладь до БД |
| Product\_database/catalog/vievs | KitchenUtensilDelete | Передача даних при видаленні кухонних приладь з БД |
| Product\_database/catalog/vievs | renew\_kitchen\_utensil | Перевірка даних при оновленні інформації о кухонних приладдях |
| Product\_database/catalog/vievs | ProductCreate | Передача даних при додаванні продукту до БД |
| Product\_database/catalog/vievs | ProductDelete | Передача даних при видаленні продукту з БД |
| Product\_database/catalog/vievs | StoreCreate | Передача даних при додаванні магазину до БД |
| Product\_database/catalog/vievs | StoreDelete | Передача даних при видаленні магазину з БД |
| Product\_database/catalog/vievs | PriceCreate | Передача даних при додаванні ціни до БД |
| Product\_database/catalog/vievs | PriceDelete | Передача даних при видалені ціни з БД |
| Product\_database/catalog/vievs | form\_valid | Перевірка форми на коректність даних |
| Product\_database/catalog/vievs | get\_queryset | Фільтр запиту до БД |
| Product\_database/catalog/tests/test\_forms | test\_renew\_form\_name\_field\_labe | Тест форми зміни інформації о продукті на коректне відображення назви |
| Product\_database/catalog/tests/test\_forms | test\_renew\_form\_name\_field\_help\_text | Тест форми зміни інформації о продукті на коректне відображення підказки |
| Product\_database/catalog/tests/test\_models | setUpTestData | Встановлення тестових даних |
| Product\_database/catalog/tests/test\_models | test\_name\_label | Перевірка відображення текстового поля імені на сайті |
| Product\_database/catalog/tests/test\_models | test\_weight\_label | Перевірка відображення текстового поля ваги на сайті |
| Product\_database/catalog/tests/test\_models | test\_name\_max\_length | Перевірка дозволеної максимальної довжини назви |
| Product\_database/catalog/tests/test\_models | test\_get\_absolute\_url | Перевірка правильно наданого посилання |
| Product\_database/catalog/tests/test\_views | test\_view\_url\_exists\_at\_desired\_location | Перевірка правильно наданого посилання |
| Product\_database/catalog/tests/test\_views | test\_view\_url\_accessible\_by\_name | Перевірка правильно наданого посилання |
| Product\_database/catalog/tests/test\_views | test\_view\_uses\_correct\_template | Перевірка на правильно підключений шаблон сторінки |
| Product\_database/catalog/tests/test\_views | test\_pagination\_is\_twenty | Перевірка на правильно виконану пагінацію сторінок |
| Product\_database/catalog/tests/test\_views | test\_lists\_all\_products | Перевірка на вірне відображення списку усіх продуктів |

Таблиця 2.3 – Таблиця опису бази даних

|  |  |  |  |
| --- | --- | --- | --- |
| Таблиця | Запис | Тип змінної | Опис |
| NutritionData | proteins | FloatField | Білки |
| NutritionData | proteins\_including\_animals | FloatField | Білки включаючи тваринні |
| NutritionData | fat | FloatField | Жири |
| NutritionData | fat\_including\_animals | FloatField | Жири включаючи рослинні |
| NutritionData | digestible\_carbohydrates | FloatField | Засвоювані вуглеводи |
| NutritionData | digestible\_carbohydrates\_incl\_m\_and\_d | FloatField | Засвоювані вуглеводи у тому числі моно- та дисахариди |
| NutritionData | dietary\_fiber | FloatField | Харчові волокна |
| NutritionData | dietary\_fiber\_including\_fiber\_and\_pectin | FloatField | Харчові волокна у тому числі клітковини та пектину |
| NutritionData | polyunsaturated\_acid | FloatField | Полі ненасичені кислоти |
| NutritionData | saturated\_acid | FloatField | Насичені кислоти |
| NutritionData | monounsaturated\_acid | FloatField | Мононасичені кислоти |
| NutritionData | calcium\_in\_mg | FloatField | Кальцій |
| NutritionData | phosphorus\_in\_mg | FloatField | Фосфор |
| NutritionData | magnesium\_in\_mg | FloatField | Магній |
| NutritionData | potassium\_in\_mg | FloatField | Калій |
| NutritionData | sodium\_in\_mg | FloatField | Натрій |
| NutritionData | chlorine\_in\_mg | FloatField | Хлор |
| NutritionData | sulfur\_in\_mg | FloatField | Сірка |
| NutritionData | iron\_in\_mg | FloatField | Залізо |
| NutritionData | zinc\_in\_mg | FloatField | Цинк |
| NutritionData | iodine\_in\_mg | FloatField | Йод |
| NutritionData | fluorine\_in\_mg | FloatField | Фтор |
| NutritionData | thiamine\_vitamin\_B1\_in\_mg | FloatField | Тіамін (В1) |
| NutritionData | riboflavin\_vitamin\_B2\_in\_mg | FloatField | Рибофлавін (В2) |
| NutritionData | pyridoxine\_vitamin\_B6\_in\_mg | FloatField | Піридоксин (В6) |
| NutritionData | pantothenic\_acid\_vitamin\_B3\_in\_mg | FloatField | Пантотенова кислота (В3) |
| NutritionData | folacin\_acid\_vitamin\_B9\_in\_mcg | FloatField | Фолацин (В9) |
| NutritionData | cobalamin\_acid\_vitamin\_B12\_in\_mcg | FloatField | Кобаломін (В12) |
| NutritionData | niacin\_vitamin\_PP\_in\_mg | FloatField | Ніацин (РР) |
| NutritionData | ascorbic\_acid\_vitamin\_C\_in\_mg | FloatField | Аскорбінова кислота (С) |
| NutritionData | retinol\_vitamin\_A\_in\_mcg | FloatField | Ретінол (А) |
| NutritionData | tocopherol\_vitamin\_E\_in\_mg | FloatField | Токоферол (Е) |
| NutritionData | cholecalciferol\_vitamin\_D\_in\_mcg | FloatField | Холекальциферол (D) |
| NutritionData | energy\_value\_in\_kcal | FloatField | Енергетична цінність |
| MenuForMultipleDays | list\_of\_menus | ManyToManyField('MenuForOneDay') | Список меню на кожен день |
| MenuForOneDay | breakfast | ForeignKey(‘Dish’) | Сніданок |
| MenuForOneDay | dinner | ForeignKey(‘Dish’) | Обід |
| MenuForOneDay | evening\_meal | ForeignKey(‘Dish’) | Вечеря |
| BalancedNutritionFormula | country | CharField(max\_length=255) | Країна |
| BalancedNutritionFormula | humans\_attributes | ManyToManyField('HumanAttributes') | Список формул для підрахунку меню збалансованого харчування |
| HumanAttributes | age | CharField(max\_length=2, choices=ALL\_YEARS) | Вік |
| HumanAttributes | sex | CharField(max\_length=1, choices=SEXES) | Стать |
| HumanAttributes | pag | CharField(max\_length=4, choices=PAGS) | Група навантаження |
| HumanAttributes | cpa | FloatField | Коефіцієнт фізичної активності |
| HumanAttributes | weigft | FloatField | вага |
| HumanAttributes | proteins | FloatField | Білки |
| HumanAttributes | proteins\_including\_animals | FloatField | Білки включаючи тваринні |
| HumanAttributes | fat | FloatField | Жири |
| HumanAttributes | fat\_including\_animals | FloatField | Жири включаючи рослинні |
| HumanAttributes | digestible\_carbohydrates | FloatField | Засвоювані вуглеводи |
| HumanAttributes | digestible\_carbohydrates\_incl\_m\_and\_d | FloatField | Засвоювані вуглеводи у тому числі моно- та дисахариди |
| HumanAttributes | dietary\_fiber | FloatField | Харчові волокна |
| HumanAttributes | dietary\_fiber\_including\_fiber\_and\_pectin | FloatField | Харчові волокна у тому числі клітковини та пектину |
| HumanAttributes | polyunsaturated\_acid | FloatField | Поліненасичені кислоти |
| HumanAttributes | saturated\_acid | FloatField | Насичені кислоти |
| HumanAttributes | monounsaturated\_acid | FloatField | Мононасичені кислоти |
| HumanAttributes | calcium\_in\_mg | FloatField | Кальцій |
| HumanAttributes | phosphorus\_in\_mg | FloatField | Фосфор |
| HumanAttributes | magnesium\_in\_mg | FloatField | Магній |
| HumanAttributes | potassium\_in\_mg | FloatField | Калій |
| HumanAttributes | sodium\_in\_mg | FloatField | Натрій |
| HumanAttributes | chlorine\_in\_mg | FloatField | Хлор |
| HumanAttributes | sulfur\_in\_mg | FloatField | Сірка |
| HumanAttributes | iron\_in\_mg | FloatField | Залізо |
| HumanAttributes | zinc\_in\_mg | FloatField | Цинк |
| HumanAttributes | iodine\_in\_mg | FloatField | Йод |
| HumanAttributes | fluorine\_in\_mg | FloatField | Фтор |
| HumanAttributes | thiamine\_vitamin\_B1\_in\_mg | FloatField | Тіамін (В1) |
| HumanAttributes | riboflavin\_vitamin\_B2\_in\_mg | FloatField | Рибофлавін (В2) |
| HumanAttributes | pyridoxine\_vitamin\_B6\_in\_mg | FloatField | Піридоксин (В6) |
| HumanAttributes | pantothenic\_acid\_vitamin\_B3\_in\_mg | FloatField | Пантотенова кислота (В3) |
| HumanAttributes | folacin\_acid\_vitamin\_B9\_in\_mcg | FloatField | Фолацин (В9) |
| HumanAttributes | cobalamin\_acid\_vitamin\_B12\_in\_mcg | FloatField | Кобаломін (В12) |
| HumanAttributes | niacin\_vitamin\_PP\_in\_mg | FloatField | Ніацин (РР) |
| HumanAttributes | ascorbic\_acid\_vitamin\_C\_in\_mg | FloatField | Аскорбінова кислота (С) |
| HumanAttributes | retinol\_vitamin\_A\_in\_mcg | FloatField | Ретінол (А) |
| HumanAttributes | tocopherol\_vitamin\_E\_in\_mg | FloatField | Токоферол (Е) |
| HumanAttributes | cholecalciferol\_vitamin\_D\_in\_mcg | FloatField | Холекальциферол (D) |
| HumanAttributes | energy\_value\_in\_kcal | FloatField | Енергетична цінність |
| HumanAttributes | related\_model | ForeignKey('BalancedNutritionFormula') | Співвідношення з моделлю формула збалансованого харчування |
| Profile | user | OneToOneField | Співвідношення з моделлю користувач |
| Profile | date\_of\_birth | DateField | Дата народження |
| Profile | sex | CharField(max\_length=1, choices=SEXES) | Стать |
| Profile | weight | FloatField | Вага |
| Profile | nursing | BooleanField | Чи виконує користувач грудне годування |
| Profile | cpa | CharField(max\_length=3, choices=PAGS) | Категорія з фізичного навантаження |
| Profile | country | CharField(max\_length=255) | Країна проживання |
| Profile | city | CharField(max\_length=255) | Місто проживання |
| Profile | location | PlainLocationField | Точне місцезнаходження яке автоматично заповнює попередні два поля |
| Product | name | CharField(max\_length=200) | Назва продукту |
| Product | favorite | ManyToManyField(User) | Список користувачів яким сподобався цей продукт |
| Product | who\_added | ForeignKey(User) | Який користувач додав цей продукт |
| Product | weight | FloatField | Вага продукту |
| Product | unit | CharField(max\_length=10, choices=UNITS) | Одиниця вимірювання ваги |
| Product | product\_by\_weight\_in\_packaging | CharField(max\_length=3) | Вага продукту в упаковці |
| Product | proteins\_in\_100\_g | FloatField | Білки в 100 г продукту |
| Product | proteins\_in\_100\_g\_including\_animals | FloatField | Білки включаючи тваринні в 100 г продукту |
| Product | fat\_in\_100\_g | FloatField | Жири в 100 г продукту |
| Product | fat\_in\_100\_g\_including\_animals | FloatField | Жири включаючи рослинні в 100 г продукту |
| Product | digestible\_carbohydrates\_in\_100\_g | FloatField | Засвоювані вуглеводи в 100 г продукту |
| Product | digestible\_carbohydrates\_in\_100\_g\_incl\_m\_and\_d | FloatField | Засвоювані вуглеводи у тому числі моно- та дисахариди в 100 г продукту |
| Product | dietary\_fiber\_in\_100\_g | FloatField | Харчові волокна в 100 г продукту |
| Product | dietary\_fiber\_in\_100\_g\_including\_fiber\_and\_pectin | FloatField | Харчові волокна у тому числі клітковини та пектину в 100 г продукту |
| Product | polyunsaturated\_acid | FloatField | Поліненасичені кислоти |
| Product | saturated\_acid | FloatField | Насичені кислоти |
| Product | monounsaturated\_acid | FloatField | Мононасичені кислоти |
| Product | calcium\_in\_100\_g\_in\_mg | FloatField | Кальцій в 100 г продукту |
| Product | phosphorus\_in\_100\_g\_in\_mg | FloatField | Фосфор в 100 г продукту |
| Product | magnesium\_in\_100\_g\_in\_mg | FloatField | Магній в 100 г продукту |
| Product | potassium\_in\_100\_g\_in\_mg | FloatField | Калій в 100 г продукту |
| Product | sodium\_in\_100\_g\_in\_mg | FloatField | Натрій в 100 г продукту |
| Product | chlorine\_in\_100\_g\_in\_mg | FloatField | Хлор в 100 г продукту |
| Product | sulfur\_in\_100\_g\_in\_mg | FloatField | Сірка в 100 г продукту |
| Product | iron\_in\_100\_g\_in\_mg | FloatField | Залізо в 100 г продукту |
| Product | zinc\_in\_100\_g\_in\_mg | FloatField | Цинк в 100 г продукту |
| Product | iodine\_in\_100\_g\_in\_mg | FloatField | Йод в 100 г продукту |
| Product | fluorine\_in\_100\_g\_in\_mg | FloatField | Фтор в 100 г продукту |
| Product | thiamine\_vitamin\_B1\_in\_100\_g\_in\_mg | FloatField | Тіамін (В1) в 100 г продукту |
| Product | riboflavin\_vitamin\_B2\_in\_100\_g\_in\_mg | FloatField | Рибофлавін (В2) в 100 г продукту |
| Product | pyridoxine\_vitamin\_B6\_in\_100\_g\_in\_mg | FloatField | Піридоксин (В6) в 100 г продукту |
| Product | pantothenic\_acid\_vitamin\_B3\_in\_100\_g\_in\_mg | FloatField | Пантотенова кислота (В3) в 100 г продукту |
| Product | folacin\_acid\_vitamin\_B9\_in\_100\_g\_in\_mcg | FloatField | Фолацин (В9) в 100 г продукту |
| Product | cobalamin\_acid\_vitamin\_B12\_in\_100\_g\_in\_mcg | FloatField | Кобаломін (В12) в 100 г продукту |
| Product | niacin\_vitamin\_PP\_in\_100\_g\_in\_mg | FloatField | Ніацин (РР) в 100 г продукту |
| Product | ascorbic\_acid\_vitamin\_C\_in\_100\_g\_in\_mg | FloatField | Аскорбінова кислота (С) в 100 г продукту |
| Product | retinol\_vitamin\_A\_in\_100\_g\_in\_mcg | FloatField | Ретінол (А) в 100 г продукту |
| Product | tocopherol\_vitamin\_E\_in\_100\_g\_in\_mg | FloatField | Токоферол (Е) в 100 г продукту |
| Product | cholecalciferol\_vitamin\_in\_100\_g\_D\_in\_mcg | FloatField | Холекальциферол (D) в 100 г продукту |
| Product | energy\_value\_in\_kcal | FloatField | Енергетична цінність |
| Store | name | CharField(max\_length=50) | Назва магазину |
| Store | country | CharField(max\_length=255) | Країна в якій знаходиться магазин |
| Store | city | CharField(max\_length=255) | Місто в якому знаходиться магазин |
| Store | street | CharField(max\_length=255) | Вулиця на якій знаходиться магазин |
| Store | location | PlainLocationField | Точне місцезнаходження магазину яке автоматично заповнює три попередні поля |
| Store | who\_added | ForeignKey(User) | Хто додав цей магазин до БД |
| Price | store | ForeignKey('Store') | Назва магазину в якому є продукт з такою ціною |
| Price | product | ForeignKey('Product') | Назва продукту на який ставиться ціна |
| Price | price | DecimalField | Ціна на вказаний продукт |
| Price | who\_added | ForeignKey(User) | Хто додав ціну на продукт |
| ProductAmount | product | ForeignKey('Product') | Продукт до якого додається його кількість |
| ProductAmount | amount | DecimalField | Вага продукту |
| ProductAmount | unit | CharField(max\_length=10) | Одиниця вимірювання ваги продукту |
| ProductAmount | related\_model | ForeignKey('Dish') | Співвідношення з блюдом до якого додавалась кількість продукту |
| KitchenUtensil | name | CharField(max\_length=100) | Назва кухонного приладдя |
| KitchenUtensil | who\_added | ForeignKey(User) | Хто додав кухонне приладдя до БД |
| Dish | name | CharField(max\_length=200) | Назва блюда |
| Dish | recipe | TextField | Рецепт блюда |
| Dish | kitchen\_utensils | ManyToManyField('KitchenUtensil') | Потрібні кухонні приладдя |
| Dish | products | ManyToManyField('ProductAmount') | Потрібна кількість продуктів |
| Dish | who\_added | ForeignKey(User) | Хто додав блюдо до БД |
| Dish | favorite | ManyToManyField(User) | Список тих кому сподобалось блюдо |

## Аналіз безпеки даних

## Дані — це найцінніший корпоративний актив для будь-якого бізнесу. Незалежно від того, в якій галузі працює підприємство, важливо дбати про фінансові звіти та медичні записи або бізнес-плани для стартапу. БД — це структурована сукупність інформації, яку можна зберігати, аналізувати та обробляти за допомогою СУБД. Бази даних необхідно захищати та регулярно перевіряти актуальність цього захисту. Використовуючи спеціальні програми та методики, можна запобігти несанкціонованому доступу (НСД) до бази даних в локальних мережах або витоку інформації, не призначеної для широкого розголосу.

Django має методи які забезпечують захист від розповсюджених різновидів атак враховуючи XSS та CSRF, захист від SQL ін’єкцій, захист від клікджекінга. Також Django має такі методи захисту як SECURE PROXY SSL HEADER, SECURE SSL REDIRECT, HTTP Strict Transport Security, SESSION COOKIE SECURE, CSRF COOKIE SECURE, валідація заголовків Host.

## Висновки до розділу

В даному розділі було детально спроектовано та описано архітектуру застосунка. В якості середовища розробки обрано Django. У якості БД використано MySQL. Розроблено всі сутності бази даних та встановлено між ними зв’язки. Для безпеки даних використано вбудовані у Django можливості.

# АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Аналіз якості ПЗ

Загалом розрізняють два базових методи тестування – мануальне та автоматизоване.

При ручному тестуванні тестери вручну виконують тести, не використовуючи ніяких засобів автоматизації. Ручне тестування – простіший тип тестування, що не вимагає великої кількості додаткових знань.

Тим не менш, перед тим як автоматизувати тестування будь-якого додатку, необхідно спочатку виконати серію тестів вручну. Мануальне тестування вимагає більших зусиль, але без нього ми не зможемо переконатися в тому, чи можлива автоматизація взагалі. Один із фундаментальних принципів тестування свідчить: 100% автоматизація неможлива. Тому, ручне тестування – це необхідність.

Для отримання максимальної користі від модульного тестування необхідно чітко слідкувати за технологією тестування протягом усього життєвого циклу розробки. Необхідно зберігати записи про всі проведенні тести та про всі зміни вихідного коду. В результаті, якщо більш пізня версія програми не проходить конкретний тест, можна звернутись до записів і перевірити на якій версії цей тест виконувався і які зміни були внесені, тобто виявити зміни, що привели до помилки і легко усунути її. Якщо ігнорувати ці вимоги, це призведе до накопичення інформації, яку вже неможливо перевірити, а тому й виправити.

Автоматизоване тестування програмного забезпечення – частина процесу тестування на етапі контролю якості в процесі розробки програмного забезпечення. Воно використовує програмні засоби для виконання тестів і перевірки результатів виконання, що допомагає скоротити час тестування і спростити його процес.

Існує два основних підходи до автоматизації тестування: тестування на рівні коду і GUI-тестування. До першого типу належить, зокрема, модульне тестування. До другого – імітація дій користувача за допомогою спеціальних тестових фреймворків.

Модульне тестування – це метод тестування програмного забезпечення, який полягає в окремому тестуванні кожного модуля коду програми. Модулем називають найменшу частину програми, яку може бути протестованою.

У процедурному програмуванні модулем вважають окрему функцію або процедуру. В об'єктно-орієнтованому програмуванні – інтерфейс, клас. Модульні тести, або unit-тести, розробляються в процесі розробки програмістами та, іноді, тестерами білої скриньки.

Інтеграційне тестування є типом тестування ПЗ, яке прагне перевірити інтерфейси між компонентами від програмного дизайну. Програмні компоненти можуть бути інтегровані як у рамках ітеративного підходу, так і всі разом.

Інтеграційне тестування працює над виявленням дефектів у інтерфейсах та взаємодії інтегрованих компонентів (модулів). Воно проводиться до тих пір, поки великі групи протестованих компонентів ПЗ, які відповідають потрібній архітектурі, починають працювати як система.

## Опис процесів тестування

Для тестування проекту було вибрано мануальні та автоматичні тести. Мануальні через економію часу на розробку. В той час як автоматичні тести використовуються або для самих важливих речей, або для тих речей котрі критичні для проекту і де треба виключити людський фактор з тестування.

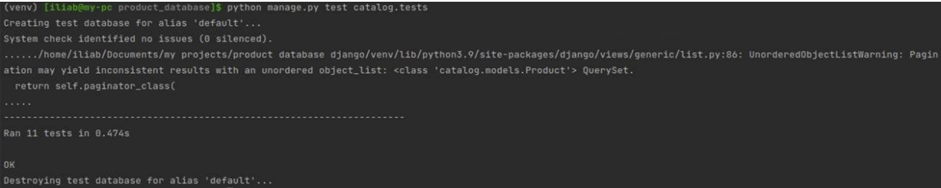


Рисунок 3.1 – Приклад запуску автономних тестів

## Опис контрольного прикладу

В таблицях 3.1 – 3.11 детально описані контрольні приклади тестування різних складових додатку.

Таблиця 3.1 – Перевірка поля name

|  |  |
| --- | --- |
| Мета тесту | Перевірка коректного відображення лейбла назви продукту на формі змінення даних продукту |
| Початковий стан | Пуста форма зміни інформації о продукті |
| Схема проведення тесту | Створюємо пусту форму та перевіряємо завдяки assertTrue поле name на формі |
| Очікуваний результат | Застосунок повинен відобразити name як в полі яке належить до поля name на формі |

Таблиця 3.2 – перевірка підказки до поля name

|  |  |
| --- | --- |
| Мета тесту | Перевірка коректного відображення підказки до поля назва продукту на формі змінення даних продукту |
| Початковий стан | Пуста форма зміни інформації о продукті |
| Схема проведення тесту | Створюємо пусту форму та перевіряємо завдяки assertEqual допоміжний текст до поля name на формі |
| Очікуваний результат | Застосунок повинен відобразити Enter product name напроти поля name на формі |

Таблиця 3.3 – перевірка поля name

|  |  |
| --- | --- |
| Мета тесту | Перевірка коректного відображення лейбла назви продукту на формі для моделі продукт |
| Початковий стан | Створюємо приблизну модель продукту |
| Схема проведення тесту | Отримуємо об’єкт продукту який був попередньо створений, отримуємо лейбл поля name та за допомогою assertEquals порівнюємо з очікуваним результатом |
| Очікуваний результат | Застосунок повинен відобразити лейбл поля name який дорівнює name |

Таблиця 3.4 – перевірка поля weight

|  |  |
| --- | --- |
| Мета тесту | Перевірка коректного відображення лейбла ваги продукту на формі для моделі продукт |
| Початковий стан | Створюємо приблизну модель продукту |
| Схема проведення тесту | Отримуємо об’єкт продукту який був попередньо створений, отримуємо лейбл поля weight та за допомогою assertEquals порівнюємо з очікуваним результатом |
| Очікуваний результат | Застосунок повинен відобразити лейбл поля weight який дорівнює weight |

Таблиця 3.5 – перевірка максимальної довжини поля name

|  |  |
| --- | --- |
| Мета тесту | Перевірка коректного показнику довжини поля назва продукту |
| Початковий стан | Створюємо приблизну модель продукту |
| Схема проведення тесту | Отримуємо об’єкт продукту який був попередньо створений, отримуємо максимальну довжину поля name та за допомогою assertEquals порівнюємо з очікуваним результатом |
| Очікуваний результат | Попередньо створена модель для тесту має мати максимальну довжину поля назва продукту рівну 200 |

Таблиця 3.6 – перевірка отримання коректного посилання на сторінку з детальною інформацією

|  |  |
| --- | --- |
| Мета тесту | Перевірка коректного створення посилання на сторінку з детальною інформацією о продукті |
| Початковий стан | Створюємо приблизну модель продукту |
| Схема проведення тесту | Отримуємо об’єкт продукту який був попередньо створений, за допомогою assertEquals порівнюємо посилання яке згенерувалося само з очікуваним результатом |
| Очікуваний результат | Застосунок повинен згенерувати посилання на сторінку з детальною інформацією о продукті яка дорівнює /catalog/product/1 |

Таблиця 3.7 – перевірка чи посилання до каталогу продуктів існує

|  |  |
| --- | --- |
| Мета тесту | Перевірка на те чи існує сторінка з списком усіх продуктів |
| Початковий стан | Створюємо 23 приблизні моделі продукту |
| Схема проведення тесту | Отримуємо відповідь від сервера при спробі потрапити на сторінку /catalog/products/, за допомогою assertEquals порівнюємо посилання яке згенерувалося само з очікуваним результатом |
| Очікуваний результат | Застосунок повинен згенерувати сторінку з списком продуктів і при спробі отримати результат цього створення відповідь від серверу повинна мати статус коду який дорівнює 200 |

Таблиця 3.8 – перевірка чи можливо отримати посилання до сторінки з детальною інформацією о продукті за допомогою назви каталогу

|  |  |
| --- | --- |
| Мета тесту | Перевірка на те чи існує сторінка з списком усіх продуктів за допомогою короткого посилання |
| Початковий стан | Створюємо 23 приблизні моделі продукту |
| Схема проведення тесту | Отримуємо відповідь від сервера при спробі потрапити на сторінку /catalog/products/ за допомогою короткого посилання reverse('products'), за допомогою assertEquals порівнюємо посилання яке згенерувалося само з очікуваним результатом |
| Очікуваний результат | Застосунок повинен згенерувати сторінку з списком продуктів і при спробі отримати результат цього створення відповідь від серверу повинна мати статус коду який дорівнює 200 |

Таблиця 3.9 – перевірка чи використовує відображення правильний шаблон веб сторінки

|  |  |
| --- | --- |
| Мета тесту | Перевірка використання коректного шаблону сторінки з детальною інформацією о продукті |
| Початковий стан | Створюємо 23 приблизні моделі продукту |
| Схема проведення тесту | Отримуємо відповідь від сервера при спробі потрапити на сторінку /catalog/products/ за допомогою короткого посилання reverse('products'), за допомогою assertEquals порівнюємо шаблон який обрався самостійно з очікуваним результатом |
| Очікуваний результат | Застосунок повинен згенерувати сторінку з списком продуктів і при спробі отримати результат цього створення шаблон сторінки повинен буди правильним |

Таблиця 3.10 – перевірка правильної роботи пагінації

|  |  |
| --- | --- |
| Мета тесту | Перевірка коректної роботи пагінації на сторінці з списком усіх продуктів |
| Початковий стан | Створюємо 23 приблизні моделі продукту |
| Схема проведення тесту | Отримуємо відповідь від сервера при спробі потрапити на сторінку /catalog/products/ за допомогою короткого посилання reverse('products'), за допомогою assertTrue перевіряємо перевіримо те що пагінація працює і саме 20 продуктів відображаються на одній сторінці |
| Очікуваний результат | Застосунок повинен згенерувати сторінку з списком усіх продуктів та на першій сторінці списку повинно бути лише 20 продуктів |

Таблиця 3.11 – перевірка правильного відображення каталогу усіх продуктів

|  |  |
| --- | --- |
| Мета тесту | Перевірка коректного створення сторінок з списком продуктів |
| Початковий стан | Створюємо 23 приблизні моделі продукту |
| Схема проведення тесту | Отримуємо відповідь від сервера при спробі потрапити на сторінку /catalog/products/?page=2 за допомогою короткого посилання reverse('products') + '?page=2', за допомогою assertTrue перевіряємо перевіримо те що пагінація працює і саме 3 продукти відображаються на другій сторінці списку усіх продуктів |
| Очікуваний результат | Застосунок повинен згенерувати сторінку з списком усіх продуктів та на другій сторінці списку повинно бути лише 3 продукти |

## Висновки до розділу

Під час тестування програмного забезпечення були виокремлені та детально описані методики тестування, критерії їх проходження. Результатом тестування стала задовільна якість застосунку та його придатність до користування

# ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## Розгортання програмного забезпечення

Першим кроком для підготовки додатку до розгортання інсталяція Python. Останню версію Python можна завантажити на офіційному сайті [7].

Наступним кроком буде підготовка та встановлення модулів для проекту, які записані в файлі requirements.txt. Строки, записані requirements.txt, являють собою назви модулів які необхідно встановити за допомогою менеджера пакетів pip. Це виконується командою “pip install -r requirements.txt”. На рисунку 4.1 зображено результат виконання цієї команди.

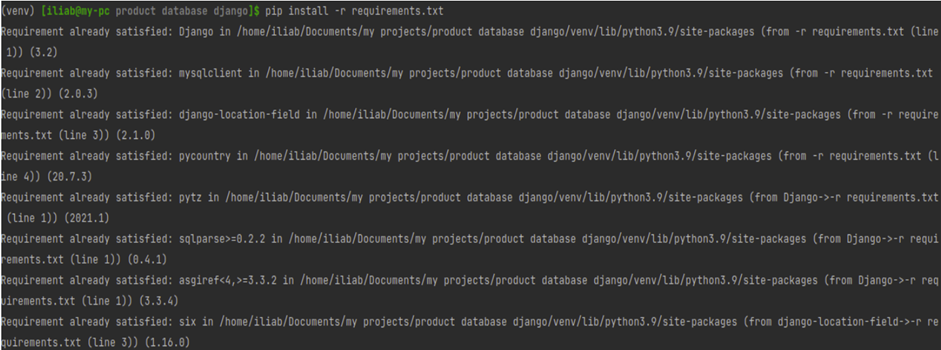


Рисунок 4.1 – Виконання команди “pip install -r requirements.txt”

Далі треба виконати “python manage.py runserver”, приклад виконання якої наведено на рисунку 4.2.

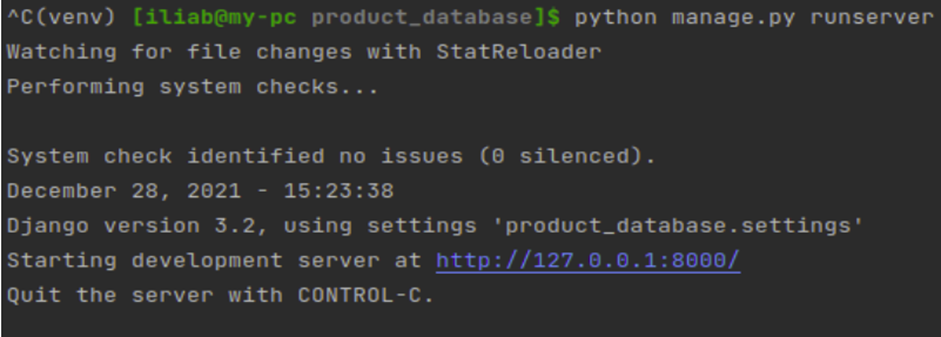


Рисунок 4.2 – Виконання команди “python manage.py runserver”

## Робота з програмним забезпеченням

Текст

### Опис основних екранів

Перше, що бачить користувач — екран реєстрації. На цьому екрані користувач має форму, після правильної заповнення якої буде створено новий обліковий запис користувача. Його вигляд зображено на рисунку 4.3.

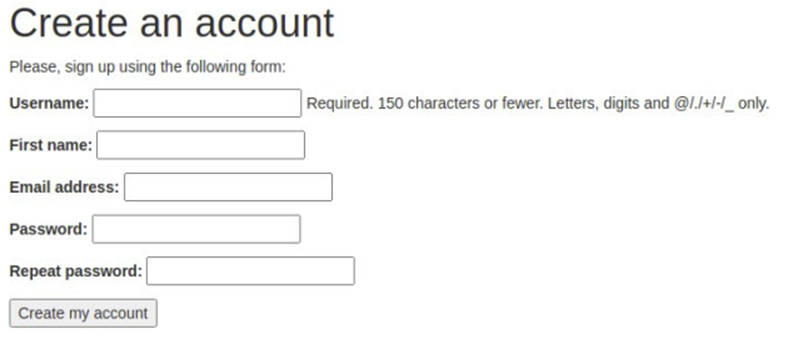


Рисунок 4.3 – Вигляд екрану реєстрації

Основним екраном додатку є домашня сторінка. На ній користувач має змогу побачити список усіх доступних йому можливостей. Вигляд цього екрану зображено на рисунку 4.4.

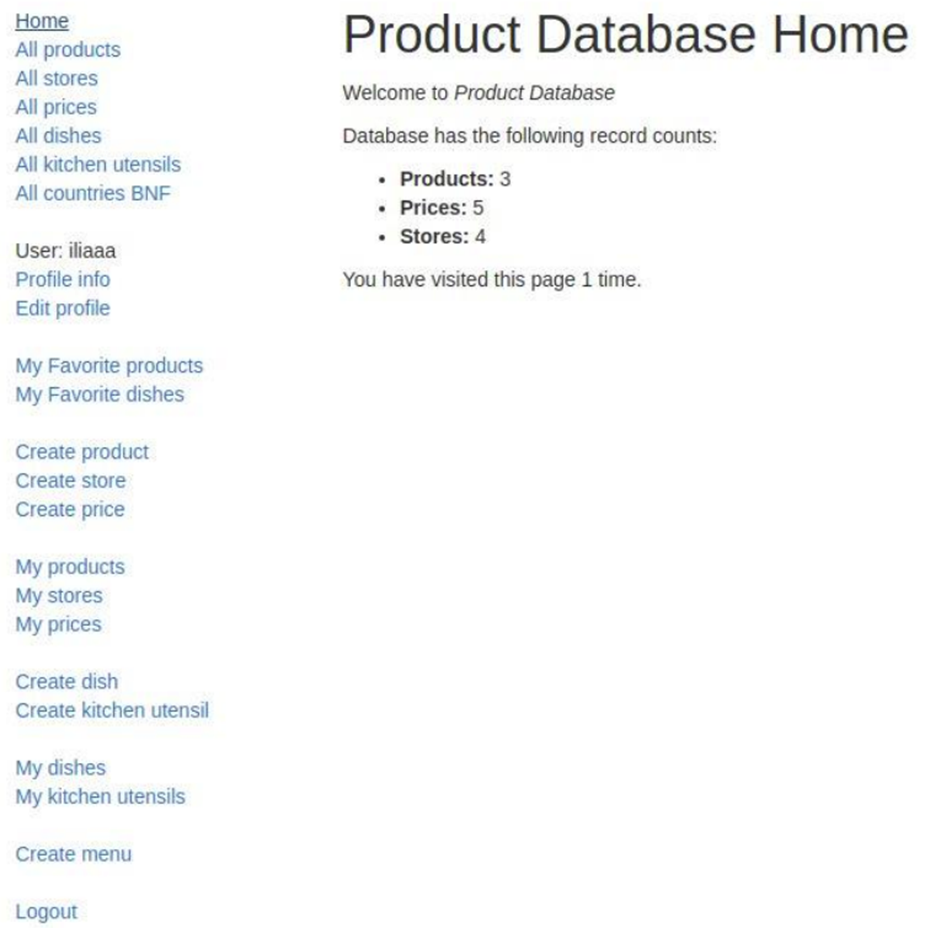


Рисунок 4.4 – Вигляд домашньої сторінки

Після того, як користувач вибирає будь-яку із доступних йому можливостей веб-додатку, його буде направлено на сторінку того що він обрав, вигляд можливих сторінок зображено на рисунках 4.5-4.12.

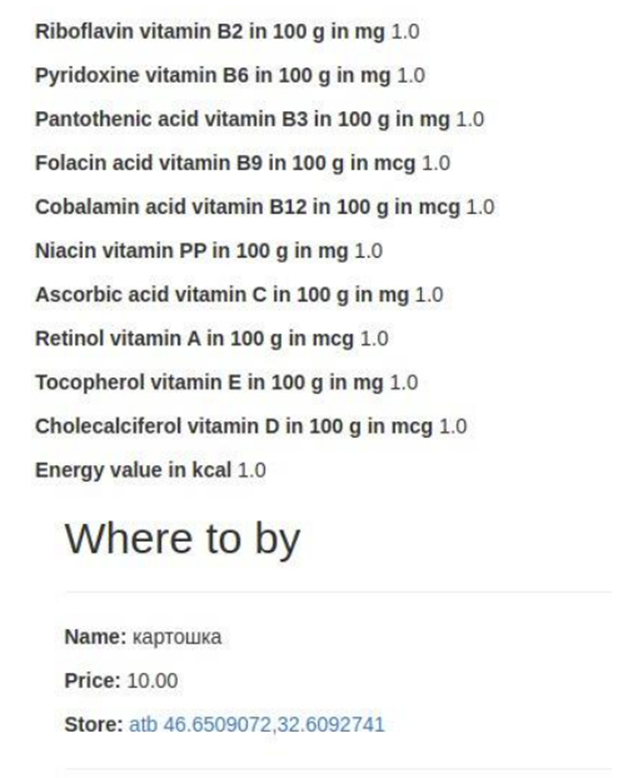


Рисунок 4.5 – Вигляд сторінки детальної інформації о продукті

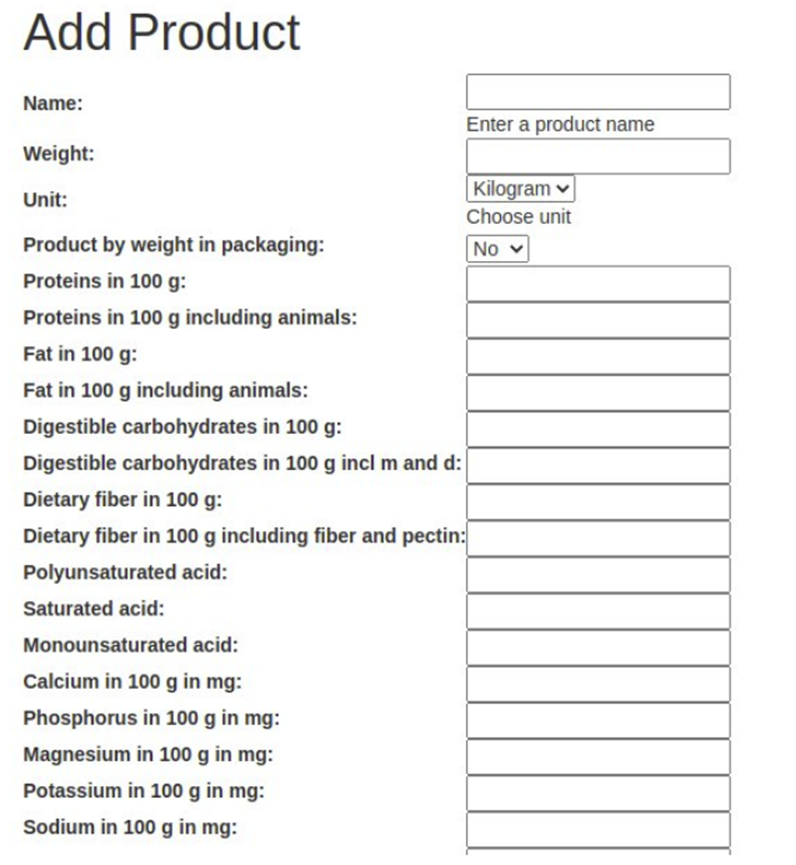


Рисунок 4.6 – Вигляд сторінки створення продукту

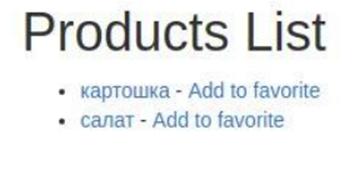


Рисунок 4.7 – Вигляд сторінки зі всіма продуктами у веб-додатку



Рисунок 4.8 – Вигляд сторінки зі всіма цінами на продукти

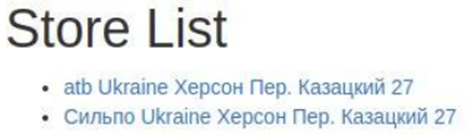


Рисунок 4.9 – Вигляд сторінки зі всіма магазинами

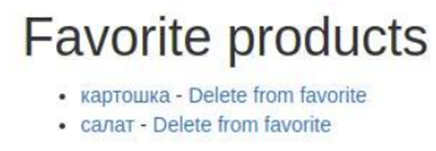


Рисунок 4.10 – Вигляд сторінки з обраними продуктами

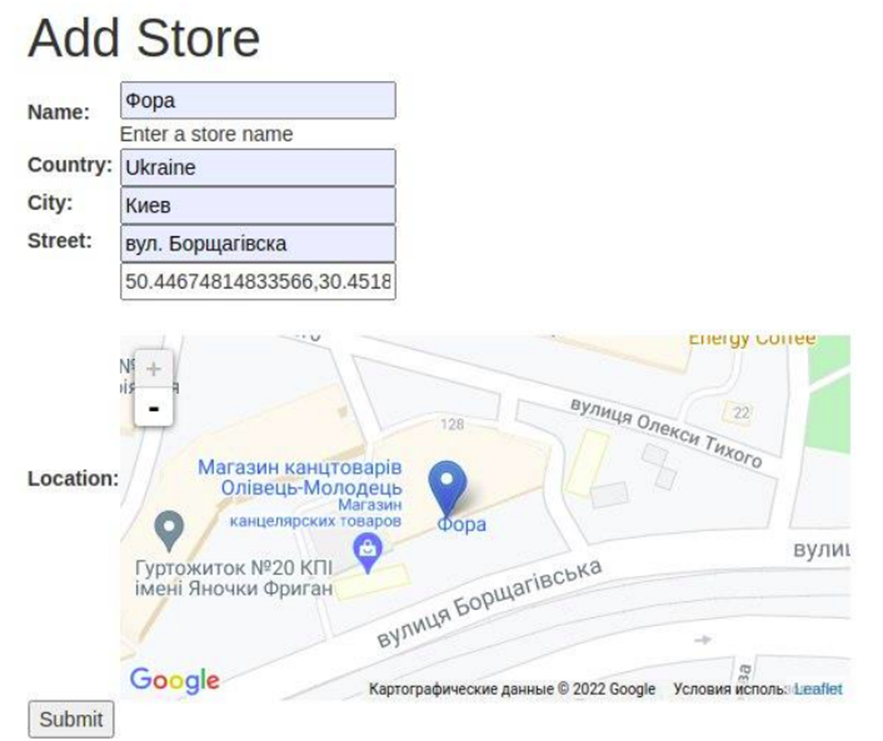


Рисунок 4.11 – Вигляд сторінки додавання супермаркету



Рисунок 4.12 – Вигляд сторінки додавання ціни на продукти

### Опис повідомлень, які виводиться користувачу

Оскільки веб-додаток містить форми для заповнення, основними повідомленнями, які буде бачити користувач під час використання веб-застосунку, будуть повідомлення помилки вводу під час заповнення форм.

Розглянемо усі типи таких повідомлень.

* «Поле не може бути пустим» — виводиться користувачу, якщо він залишив обов’язкове поле пустим.
* «Помилкова форма електронної адреси» — з’являється, якщо введений електронний адрес не відповідає стандартам адрес.
* «Недостатня кількість символів» — вказує на те, що довжина заданого логіну чи паролю є меншою за мінімальну допустиму.
* «Пароль не відповідає критеріям безпеки» — виводиться користувачу, коли введений ним пароль не відповідає одному чи більше критерію, які необхідні для створення безпечного паролю.
* «Підтвердження паролю не збігається із реальним» — означає, що поля “Підтвердження паролю” та “Пароль” не є ідентичними.

## Висновки до розділу

В даному розділі були розписані всі необхідні кроки для встановлення залежності програмного комплексу та його розгортання. Інструкції для звичайного користувача також було наведено.

# ВИСНОВКИ

В результаті виконання дипломного проекту було спроектовано та розроблено та втілено створену архітектуру програмного забезпечення. По завершенню розробки будо реалізовано план тестування та відповідно до нього протестовано реалізований програмний засіб.

Також створена детальна інструкція з підготовки середовища для розгортання та запуску програмного засобу на апаратній платформі.

Після реалізації застосунку він був протестований на пристроях з різними операційними системами, з різними розмірами екранів щоб переконатися, що додаток акуратно відображається на різних пристроях.

Було створено веб-додаток який спроможній створювати меню збалансованого харчування на ту суму і кількість днів яку захотів користувач.

Також було створено функціонал додавання та видалення таких елементів як магазини, продукти та ціни на них в різних магазинах, блюда та кухонні приладдя. Додана панель адміністратора.

В якості середовища розробки обрано Django. У якості БД використано MySQL.

Під час роботи над даним дипломним проектом були поглиблені та застосовані знання з реляційних баз даних, розробки на мові програмування Python, архітектури програмного забезпечення, розробки інтерфейсів, були вивчені нові інструменти тестування.

Результатом виконання роботи було створення повноцінного робочого програмного додатку, що готовий до використання. Важливою можливістю є використання даного алгоритму в майбутньому з допомогою впровадження меню з ресторанів для того щоб можливо було обрати наприклад що в якийсь день з меню користувач хоче не готувати їжу сам, а поїсти в ресторані. Таким чином програма може сформувати меню збалансованого харчування враховуючи блюдо яке користувач з’їсть в ресторані. Також можливо додати функцію передачі списку покупок до служби доставки продуктів. Ще однин варіант використання, це відкриття нового бізнесу який би займався приготуванням страв за рецептами наданими програмою для того щоб надати можливість людям які не можуть або не хочуть самі готувати теж харчуватися збалансовано. Одним з варіантів використання є доповнення до цього алгоритму такої можливості як підрахунок потрібної кількості їжі для певного регіону з врахуванням доступних в регіоні продуктів. Таким чином можливо вияснити чи достатньо даному регіону їжі для збалансованого харчування. Отже можливо зробити такий розрахунок для кожного регіону на Землі з врахуванням можливості транспортування продуктів з одного регіону до іншого. Виходячи з цього можливо розрахувати чи достатньо на Землі продуктів для збалансованого харчування кожної людини на планеті та зробити висновки з даної інформації, можливо потрібно збільшити кількість полів та ферм, або потрібно накопичувати продукти в місцях для схову.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. [Оформлення бібліографії ДСТУ 7.1:2006](https://drive.google.com/file/d/1VUr7fwKHOk8t1u8sh-Sv2dlEc_6OBoIS/view?usp=sharing)
2. [Оформлення бібліографії ДСТУ 8302-2015](https://drive.google.com/file/d/17RYEYVnSXolL0S8D1k85VfReYd2yC6WO/view?usp=sharing) з прикладами
3. Порівняння цін у супермаркетах України [Електронний ресурс] – Режим доступу до ресурсу: <https://price.ua/ua>
4. Додаток з таблицею калорійності [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kaloricketabulky.cz/>
5. Додаток для створення меню харчування [Електронний ресурс] – Режим доступу до ресурсу: <https://www.marysrecipes.com/ru>
6. Hunt, Andrew; Thomas, David. Pragmatic Unit Testing in Python, 2nd Ed. The Pragmatic Bookshelf (Raleigh), 2007. 378 с.
7. Офіційний сайт Python [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/>
8. Документація web-фреймворку Django [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.djangoproject.com/en/4.0>
9. Документація до стандартної бібліотеки Python [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/library/index.html>
10. MVT архітектура Django [Електронний ресурс] – Режим доступу до ресурсу: [https://www.onlinetutorialspoint.com/django/django-model-view-template-mvt-overview.html](https://www.onlinetutorialspoint.com/django/django-model-view-template-mvt-overview.html%20)

# ДОДАТКИ