

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень


Інформаційна система підбору товарів
Назва теми

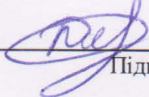
КВРІСТ.101041. 21.01.03 ПЗ
Шифр

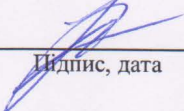
Галузь знань 12 «Інформаційні технології»
Шифр, назва

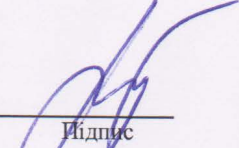
Спеціальність 126 «Інформаційні системи та технології»
Шифр, назва

Освітня програма «Інформаційні системи та технології»
Назва

Виконав: студент IV курсу, група ICTc-21-1  А. В. Кириєнко
Підпис Ініціали, прізвище

Керівник  Д. О. Денисюк
Підпис, дата Ініціали, прізвище

Нормоконтролер  І.О. Засорнова
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних систем
систем  Т.О. Говорущенко
Підпис Ініціали, прізвище

«10» червня 2024 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 11 ” 01 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Кириєнко Анатолію Володимировичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система підбору товарів

Керівник проекту (роботи) Денисюк Д.О., ст.викладач

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 15.02.2024 р. № 8

2. Строк подання студентом проекту (роботи) на кафедру 07.06.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та постановка задачі

Проектування інформаційної системи підбору товарів

Програмна реалізація інформаційної системи підбору товарів





5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Схема бази даних.

UML діаграма варіантів дій інформаційної системи підбору товару

UML діаграма варіантів дій інформаційної системи підбору товару

6. Консультанти розділів дипломного проекту (роботи)

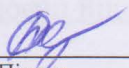
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Засорнова І.О., доцент кафедри КІСП		
Антиплагіат	Нічепорук А.О., доцент кафедри КІСП		

7. Дата видачі завдання « 11 » 01 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	11.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2024	виконано
4	Робота над розділом 2 – проектування інформаційної системи підбору товарів	01.04.2024	виконано
5	Робота над розділом 3 – програмна реалізація інформаційної системи підбору товарів	30.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	31.05.2024	виконано
7	Попередній захист ВКР	30.05.2024	виконано
8	Захист ВКР на засіданні ЕК	Червень 2024 року	

Студент


Підпис

А. В. Кириєнко
Ініціали, прізвище

Керівник проекту (роботи)


Підпис

Д. О. Денисюк
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система підбору товарів».

Автор роботи: Кириєнко Анатолій Володимирович.

Керівник роботи: Денисюк Дмитро Олександрович.

Пояснювальна записка: 70 с., 56 рис., 2 табл., 49 джерел.

Графічна частина: 3 креслення.

ІНФОРМАЦІЙНА СИСТЕМА, БАЗА ДАНИХ, АВТОМАТИЗАЦІЯ, ВЕБ ДОДАТОК, ЕЛЕКТРОННА АПТЕКА, ПІДБІР ТОВАРІВ.

Метою роботи є проектування та програмна реалізація інформаційної система підбору товарів.

Об'єктом дослідження є інформаційна система підбору товарів медичного призначення.

Предметом дослідження є предметна область для програмного комплексу (інформаційної системи) підбору товарів медичного призначення.

Практичне значення має спроектована та реалізована програмно інформаційна система підбору товарів, що використовується для реалізації можливостей пошуку, підбору та замовлення товарів медичного призначення.




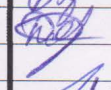
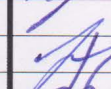
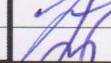
Підпис студента

10.06.2024

Дата

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	3
ВСТУП	4
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	6
1.1 Аналіз вхідних та вихідних даних інформаційної системи підбору товарів.....	6
1.2 Огляд існуючих рішень, методів та засобів розробки інформаційних систем.....	10
1.3 Формалізована модель інформаційної системи підбору товарів.....	17
1.4 Висновки. Постановка задачі.....	21
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДБОРУ ТОВАРІВ	23
2.1 Проєктування архітектури інформаційної системи підбору товарів.....	23
2.2 Проєктування бази даних інформаційної системи підбору товарів.....	28
2.3 Проєктування інтерфейсу інформаційної системи підбору товарів.....	39
2.4 Висновки.....	45
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДБОРУ ТОВАРІВ	46
3.1 Програмна реалізація веб додатку.....	46
3.2 Тестування проєкту.....	56
3.2.1. Функціональне тестування.....	56
3.2.2. GUI тестування.....	58
3.2.3. Модульне тестування.....	59
3.3 Інструкція з інсталяції та експлуатації.....	62
3.4 Висновки.....	65
ВИСНОВКИ	66
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	67

КВРІСТ.101041. 21.01.03 ПЗ								
Зм.	Арк.	№докум.	Підпис	Дата	Інформаційна система підбору товарів. Пояснювальна записка	Літера	Арквип	Арквипів
Виконав		Кириєнко А. В.		10.01		y		70
Перевір.		Денисюк Д.О.		10.06				
Н.контр.		Засорнова І.О.		10.06				
Затвер.		Говорущенко Т.О.		10.06			ХНУ, ІСТс-21-1	

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ІС – інформаційна система

АІС – автоматизована інформаційна система

БД – база даних

СУБД – система управління базами даних

НФ – нормальна форма

ІТ – інформаційні технології

АТС – анатомо-терапевтична класифікація лікарських засобів (Anatomical Therapeutic Chemical)

UML – Unified Modelling Language або універсальна мова моделювання

SQL – Structured Query Language або структурована мова запитів

HTML – мова розмітки гіпертексту (HyperText Markup Language)

CSS – спеціальна мова стилю сторінок (Cascading Style Sheets)

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Спеціалізовані комп'ютерні системи увійшли в усі сфери людського життя: пральні машини, мультиварки, телефони, органайзери, телевізори, комп'ютерні ігри, і уявити сьогодні звичайний буденний день без цього всього неможливо. Проте окрім систем, що забезпечують ефективне функціонування пристроїв, сьогодні важко уявити ще й пересічну людину, що не використовує можливості мережі Інтернет, що не є учасником соціальних мереж та спільнот, чи не користується різноманітними веб-сервісами.

У Інтернеті існують спеціальні сервіси, що дозволяють безкоштовно або за певну платню (мають більшу кількість дозволених сервісів), на певних умовах, розміщувати Інтернет сайти, форуми, магазини та інші веб-ресурси.

Сьогодні глобальна мережа Internet є не тільки інструментом обміну інформацією, вона вже давно переросла в інструмент для бізнесу у різних сферах, а саме: рекламний бізнес, інтернет комерція, надання інтернет послуг, аптечні мережі та інше. Тому дана мережа набула такого значного розвитку за такий короткий час.

«Змагання» між компаніями за ринок збуту продукції спричинило розвиток інтернет маркетингу, тобто знань та дій спрямованих на збільшення продажів за допомогою спеціалізованих інформаційних систем – веб додатків. Не винятком є й такий напрямок інтернет торгівлі, як електронна торгівля аптечних закладів.

Великою перевагою таких інформаційних систем є швидкість, зручність у підборі та замовленні клієнтами потрібних медичних препаратів, або іншої аптечної продукції.

Беручи до уваги використання комп'ютерних пристроїв у повсякденному житті людей, яким зручніше та швидше переглядати та здійснювати пошук, замовлення потрібних медичних товарів використовуючи можливості глобальної мережі.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

Часто для зацікавлення клієнтів в аптеках реалізується система знижок на товари, проводяться різноманітні акції, що також зручніше реалізувати, контролювати саме в електронному вигляді

З іншого боку аптеці, як суб'єкту комерційної діяльності зручно додавати, оновлювати, видаляти медичні товари безпосередньо за допомогою інформаційної веб системи, керувати та контролювати замовлення медтоварів клієнтами, реалізовувати доставку замовлень за допомогою поштових сервісів, організовувати акційні заходи, рекламувати певні товари.

Мета кваліфікаційної роботи полягає у розробці інформаційної системи підбору товарів, яка надасть користувачу можливість зручно здійснювати пошук потрібних позицій товару у програмній системі.

Об'єктом дослідження будуть медичні препарати, тобто товари медичного характеру, які завжди є актуальними, оскільки в час пандемій та вірусів люди постійно звертаються до аптек та медичних закладів у пошуку необхідних лікувальних засобів.

Отже, розробка інформаційної системи підбору товарів медичного призначення є актуальною задачею, має практичне значення та, потенційно, буде мати попит на ринку ІТ продуктів, насамперед, серед аптечних мереж, які намагаються перенести торгівлю також і в Інтернет.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз вхідних та вихідних даних інформаційної системи підбору товарів

Дослідження та аналіз джерел показали, що інформаційні системи підбору товарів – це програмні рішення, які допомагають користувачам знайти і підібрати необхідний товар серед великого асортименту. Такі рішення часто використовуються у інтернет-магазинах для полегшення пошуку та вибору товарів клієнтами. Працюють на основі фільтрів, які дозволяють відсіювати непотрібні варіанти за певними критеріями (ціна, бренд, характеристики, наявність товару тощо).

Такі інформаційні системи допомагають збільшити конверсію і продажі за рахунок полегшення процесу вибору товару для користувача.

Отже, це корисні інструменти, які оптимізують онлайн-шопінг і підвищують лояльність клієнтів за рахунок персоналізації.

Також існує різновид інформаційних систем для підбору товарів медичного призначення. У загальному така система реалізується у вигляді інтернет-магазину, який продає лікарські засоби та інші товари медичного призначення онлайн. До переваг електронних аптек можна віднести: зручність (можливість замовити товари не виходячи з дому), широкий асортимент, можливість порівняти ціни в різних аптеках, знижки та акції.

Серед послуг, які надають електронні аптеки: продаж ліків за рецептами і без рецептів, доставка замовлень, консультації онлайн фармацевтів.

Для того, щоб реалізувати таку задачу слід дослідити та проаналізувати вхідні та результуючі інформаційні характеристики предметної області.

Спершу визначимось із основними вхідними інформаційними сутностями та їх характеристиками.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

В такій системі обов'язково має бути інформація про самі товари, які слід підібрати. У нашому випадку – це товари медичного призначення, що мають наступні вхідні атрибути:

- назва медичного товару;
- виробник;
- АТС код (Класифікаційна система АТС дозволяє ідентифікувати лікарський препарат, у тому числі діючу речовину, визначити спосіб його введення, а також, у разі вказаної DDD (Defined Daily Dose), добову дозу споживання);
- інструкція до використання;
- фармацевтична група товару;
- категорія товару (лікарські засоби, косметика, медтехніка) ;
- міжнародна назва товару;
- дата виготовлення;
- дата придатності;
- посилання на youtube огляд;
- категорія товару згідно АТС;
- форма випуску товару (пластина, коробка);
- одиниця вимірювання (пластина, коробка);
- країна виробник;
- діючі речовини;
- правила дозування;
- інші характеристики товару;
- зображення товару;
- аналоги товару;
- вартість товару;
- вартість отримання;
- вартість акційна (зі знижкою);

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

- наявна кількість товару.

Наступна сутність предметної області – «Користувачі» має такі характеристики:

- ПІБ (ім'я) користувача;
- електронна адреса користувача;
- пароль;
- телефон;
- місто;
- статус користувача (активний чи неактивний) ;
- дата реєстрації;
- дата дереєстрації;
- список улюблених товарів;
- відгуки користувача.

Логічно, що користувачі мають робити замовлення товарів. Тому й визначимо характеристики цього об'єкта предметної області. Ними будуть:

- користувач, який робить замовлення;
- назва замовлення;
- телефон для контактів;
- опис замовлення;
- загальна вартість замовлення;
- статус замовлення (виконано/не виконано) ;
- дата створення замовлення;
- дата виконання замовлення;
- медичний товар, що його замовляють;
- кількість товару;
- вартість товару.

Розглянувши основну вхідну інформацію, визначимо які ж будуть основні вихідні (результуючі) дані та способи їх формування.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

Описана вхідна інформація має бути оброблена в розроблюваному програмному забезпеченні й результат опрацювання повинен отримати користувач у якості вихідної інформації. Вона буде виводитись, наприклад, у вигляді каталогу товарів чи сформованого замовлення, у відповідних вікнах, вкладках інформаційної системи. Проте нижче з'ясуємо саме критерії формування вихідної інформації залежно від потреб користувача.

Однією із основних операцій інформаційної системи є підбір медичного товару для конкретного клієнта за вказаними ним критеріями. Такими критеріями будуть:

- категорія товару (лікарські засоби, косметика, медтехніка);
- вартість товару (доцільно визначити діапазон від мінімальної до максимальної ціни, що прийнятна для конкретного користувача);
- форма випуску товару (пластина, коробка);
- країна виробник;
- умови відпуску (за рецепту, без рецепту).

Вказання таких критеріїв дозволить користувачу здійснити максимально точний підбір та відобразити саме такі товари медичного призначення, що відповідають його потребам.

Для зручного виведення інформації про товари медичного призначення доцільно передбачити можливість сортування за наступними критеріями:

- від дешевих до дорогих;
- від дорогих до дешевих;
- за рейтингом.

В результаті перед користувачем будуть відображені товари медичного призначення у відповідному порядку.

Ще однією важливою функцією інформаційної системи є формування та виведення замовлення користувача. Для цього користувач має обрати потрібні товари медичного призначення та додати їх у кошик замовлення. Після

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

закінчення формування замовлення, користувач зможе побачити вікно кошика замовлення.

Це вікно має містити наступні вихідні дані:

- зображення та назву замовленого товару;
- обрана кількість товару, його ціна;
- загальна вартість.

У кошику слід додати ім'я, телефон, додаткові побажання для замовлення та підтвердити замовлення.

Таким чином буде сформоване вихідне замовлення, що буде опрацьоване відповідними працівниками для його виконання.

1.2 Огляд існуючих рішень, методів та засобів розробки інформаційних систем

Для повного розуміння задачі, крім аналізу вхідних та вихідних даних предметної області проведемо огляд та аналіз існуючих рішень, що мають аналогічне призначення.

Розглянемо найпопулярніші веб ресурси для підбору та замовлення товарів медичного призначення.

Для цього використаємо рейтинг усіх інтернет аптек України, де можна замовити ліки онлайн, що сформовано на підставі оцінок та відгуків покупців, що приведено на інтернет ресурсі <https://shoptop.kiev.ua/category/internet-apteki/>.

Спершу розглянемо інтернет ресурс <https://apteka911.ua/ua>, вигляд головної сторінки якого відображено на рисунку 1.1.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10



Рисунок 1.1 – Головна сторінка інтернет ресурсу <https://apteka911.ua/ua>

Далі розглянемо інтернет ресурс apteka24.ua, вигляд головної сторінки якого відображено на рисунку 1.2.

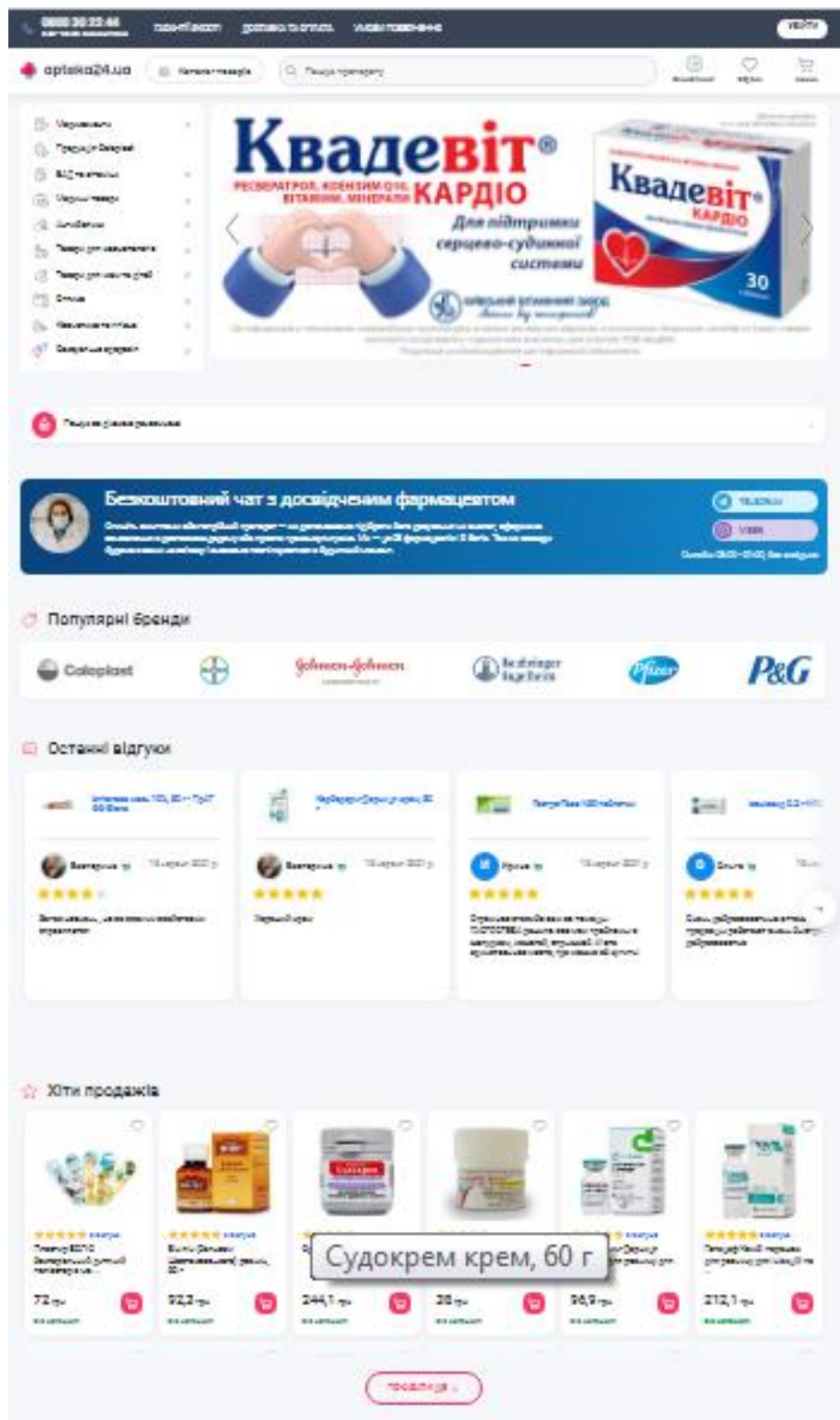


Рисунок 1.2 – Головна сторінка інтернет ресурсу arteka24.ua

Зм.	Арк.	№ докум.	Підпис	Дата

КВРІСТ.101041. 21.01.03 ПЗ

Арк.

12

Наступним розглянемо інтернет ресурс <https://tabletki.ua/>, вигляд головної сторінки якого відображено на рисунку 1.3.

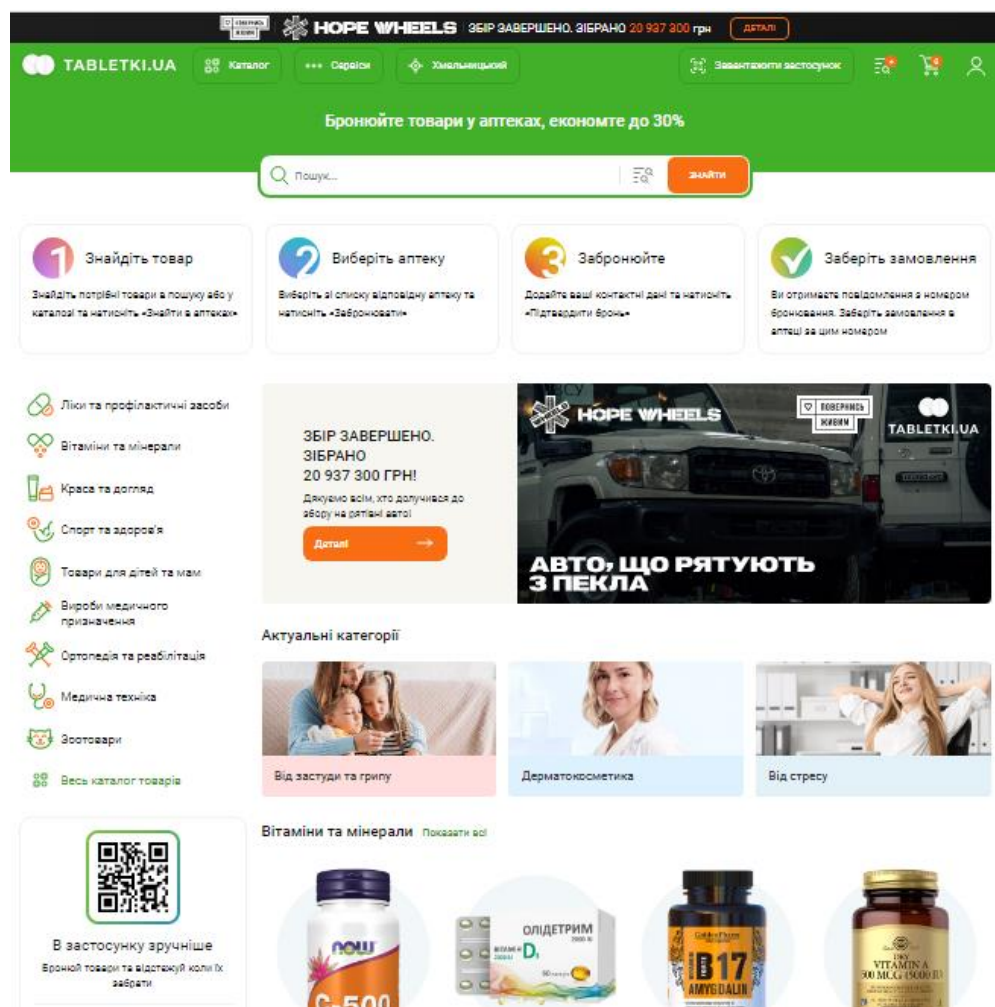


Рисунок 1.3 – Головна сторінка інтернет ресурсу <https://tabletki.ua/>

Загальний аналіз цих інтернет-аптек дозволив визначити наступні можливості, переваги та недоліки.

Можливості:

- зручність та доступність: забезпечення зручності замовлення ліків та інших товарів для здоров'я з будь-якого місця, де є доступ до мережі Інтернет;

- широкий асортимент товарів: можливість висвітлити на вебсайті широкий вибір лікарських засобів, вітамінів, дитячих товарів та інших продуктів для здоров'я;
- доставка: інтернет-сайти надають можливість користувачеві обрати оптимальний спосіб доставки замовленого товару;
- функції пошуку та фільтрації: простий та швидкий механізм пошуку продукту за необхідними критеріями;
- наявність мобільних додатків для Android та iOS, що забезпечує максимальну зручність та охоплення користувачів різних електронних пристроїв;
- пропонується система знижок для постійних клієнтів та акційні ціни.

До переваг розглянутих інтернет-аптек віднесемо:

- ефективне управління замовленнями: клієнти мають можливість відстежувати статус опрацювання свого замовлення;
- онлайн-консультації: інтернет-аптеки забезпечують можливість надання консультацій клієнтам з медичних питань використовуючи чати в месенджерах або телефонію;
- економія часу: замовлення товару за декілька кліків значно економлять час не витрачаючи його на поїздку до аптеки.

До недоліків розглянутих інтернет-аптек віднесемо:

- тривалість доставки: хоча багато інтернет-аптек пропонують доставку, у деяких випадках це може займати деякий час, особливо у великих містах;
- ризик підробки товарів: будь-який неперевірений інтернет-ресурс несе певні ризики, адже просто через мережу натрапити на шахраїв, відповідно через це існує ризик отримання підроблених або низької якості товарів;

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

Особливості інтерпретованих:

- динамічна область видимості змінних;
- менший розмір виконуваних файлів;
- динамічна типізація.

Вибір зупинився на мові програмування PHP з використанням фреймворку Laravel. Laravel – безкоштовний, з відкритим кодом PHP-фреймворк, створений Тейлор Отвел і призначений для розробки веб додатків.

Перевагами даного фреймворку є:

- повна автентифікація та авторизація користувачів;
- кросплатформова інтеграція з поштою та системами обміну повідомленнями;
- інтеграція зі швидким кешуванням;
- швидка маршрутизація URL.

Для створення візуальної частини сайту, зокрема для реалізації структури сайту було використано HTML (англ. HyperText Markup Language – мова розмітки гіпертексту) – стандартизовану мову розмітки документів для перегляду веб сторінок у браузері. для стилізації, візуального оформлення веб додатку використаємо CSS. CSS (англ. Cascading Style Sheets, укр. Каскадні таблиці стилів) – це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду.

При виборі СУБД для проекту, було розглянуто PostgreSQL та MySQL. Обидві виконують однакову функцію, хоча маю різні варіанти використання. Вони відрізняються деякими особливостями, але обидві системи базуються на SQL або Structured Query Language (структурована мова запитів). У зв'язку з цим можна виявити кілька схожостей між MySQL і PostgreSQL, таких як використання таблиць для збереження даних, посилання на первинні та зовнішні ключі, так само як кілька баз даних в одному середовищі або на одному сервері.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

Після аналізу цих двох СУБД, було вибрано MySQL, так як він краще підходить для додатків, що написані на PHP.

Для реалізації програмного забезпечення було обрано IDE PhpStorm, серед популярних середовищ розробки, таких як: Sublime Text, Visual Studio Code, WebStorm, PhpStorm та інших. Оскільки PhpStorm легке й ненавантажене середовище для розробки на різноманітних мовах, що має ряд особливостей, таких як: встановлення плагінів, що полегшують роботу з тими ж мовами (наприклад, плагін для роботи з шаблонами edge), підказки в функціях, підсвічування синтаксису та інші.

Проведена робота дозволяє перейти до проектування загальних формальних моделей інформаційної системи.

1.3 Формалізована модель інформаційної системи підбору товарів

Для побудови формалізованої моделі інформаційної системи використаємо досить потужний інструмент UML діаграм.

Універсальна мова моделювання (Unified Modelling Language або UML) – це мова позначень або побудови діаграм, призначена для визначення, візуалізації і документування моделей зорієнтованих на об'єкти систем програмного забезпечення. UML не є методом розробки, іншими словами, у конструкціях цієї мови не повідомляється про те, що робити першим, а що останнім, і не надається інструкцій щодо побудови вашої системи, але ця мова допомагає наочно переглядати компонування системи і полегшує співпрацю з іншими її розробниками. Ця мова є загальноприйнятим стандартом графічного опису програмного забезпечення.

Діаграма прецедентів (або діаграма варіантів використання) (англ. Use case diagram) – в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Діаграма прецедентів показує різні варіанти використання та різні типи користувачів системи.

Спершу спроектуємо діаграму прецедентів для звичайного користувача інформаційної системи (рисунок 1.4).

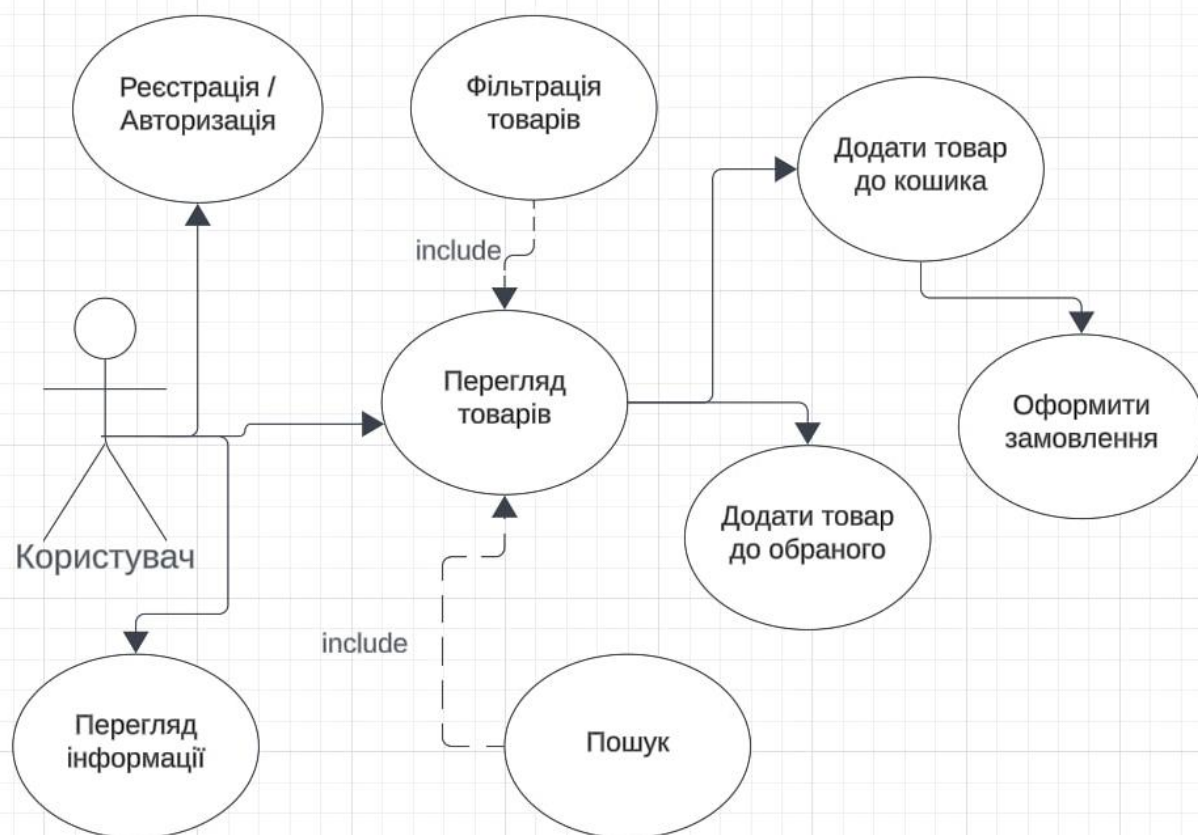


Рисунок 1.4 – Діаграма прецедентів для звичайного користувача

Отже, звичайний користувач інформаційної системи може без реєстрації/авторизацію переглядати різноманітну інформацію на сторінках веб сайту. Наприклад переглядати інформацію про товари, про сайт та контакти, про класифікацію АТС, про оплату та доставку, про акційні пропозиції тощо.

Для повноцінної роботи із інформаційною системою користувач має зареєструватись, а надалі авторизуватись. Після цього він може здійснити підбір потрібних медичних товарі за допомогою таких доступних йому інструментів як пошук та фільтрація. Відповідні критеріям пошуку, фільтрації товари, що

зацікавили користувача, він може додати в обрані, або ж у кошик. Товари з кошику формують замовлення.

Далі спроектуємо діаграму прецедентів для користувача «адміністратор» інформаційної системи (рисунок 1.5).

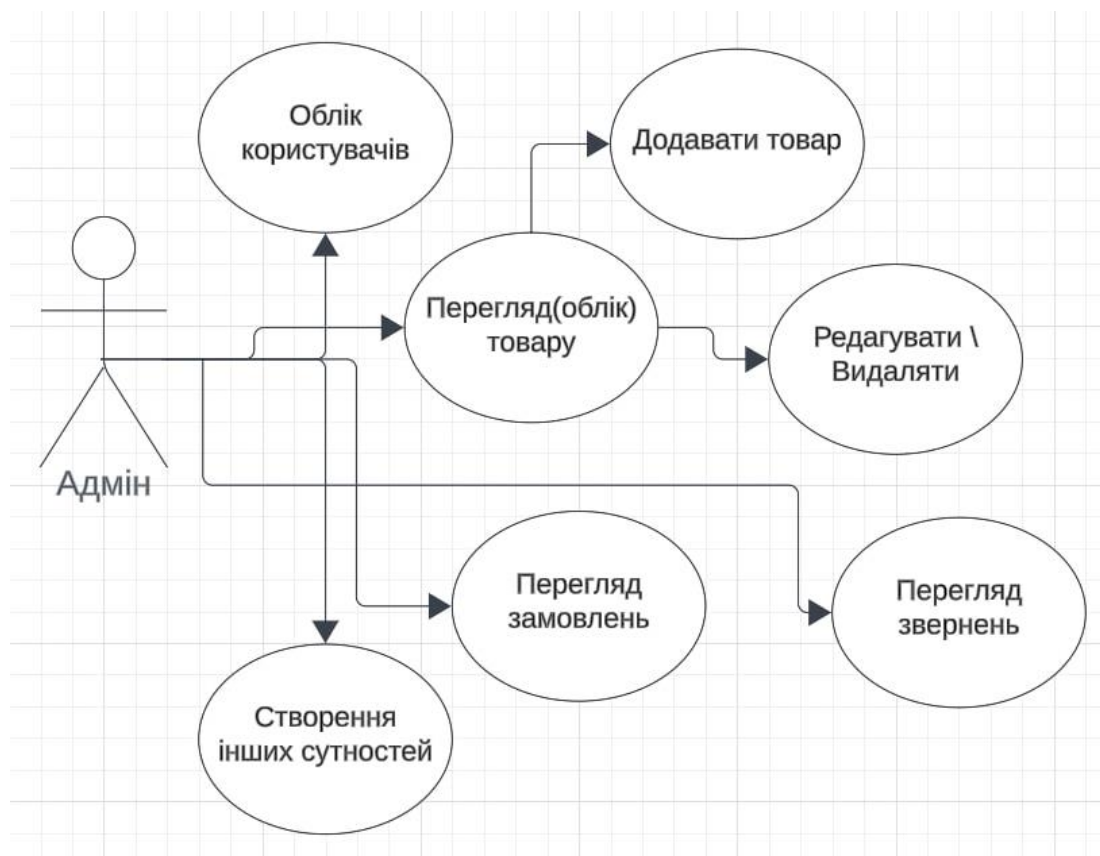


Рисунок 1.5 – Діаграма прецедентів для адміністратора

Як видно із рисунку 1.5 користувач «адміністратор» насамперед має працювати із адміністративною частиною сайту.

Зокрема він може керувати, вести облік користувачів системи.

Також йому доступний інструментарій не лише для перегляду та обліку товарів інформаційної системи, а й для додання нових товарів, редагування інформації про них або ж видалення товарів.

Зм.	Арк.	№ докум.	Підпис	Дата

Однією із важливих задач «менеджера-адміністратора» є опрацювання замовлень з метою їх підтвердження, формування, можливої доставки, контролю стану виконання та завершення замовлення, а також опрацювання звернень з метою забезпечення зворотного зв'язку з користувачами.

Для розширення можливостей сайту «адміністратор» може створювати нові сутності.

Варіанти використання інформаційної системи показані вище і частково формалізують задачу. Створимо діаграму послідовності дій для відображення порядку обміну повідомленнями при здійсненні замовлення, що є однією із діаграм взаємодії (рисунок 1.6).

Діаграма послідовності (sequence diagram) – діаграма, на якій показані взаємодії об'єктів, упорядковані за часом їхнього прояву. На діаграмі послідовності неявно присутня вісь часу, що дозволяє візуалізувати тимчасові відношення між переданими повідомленнями.

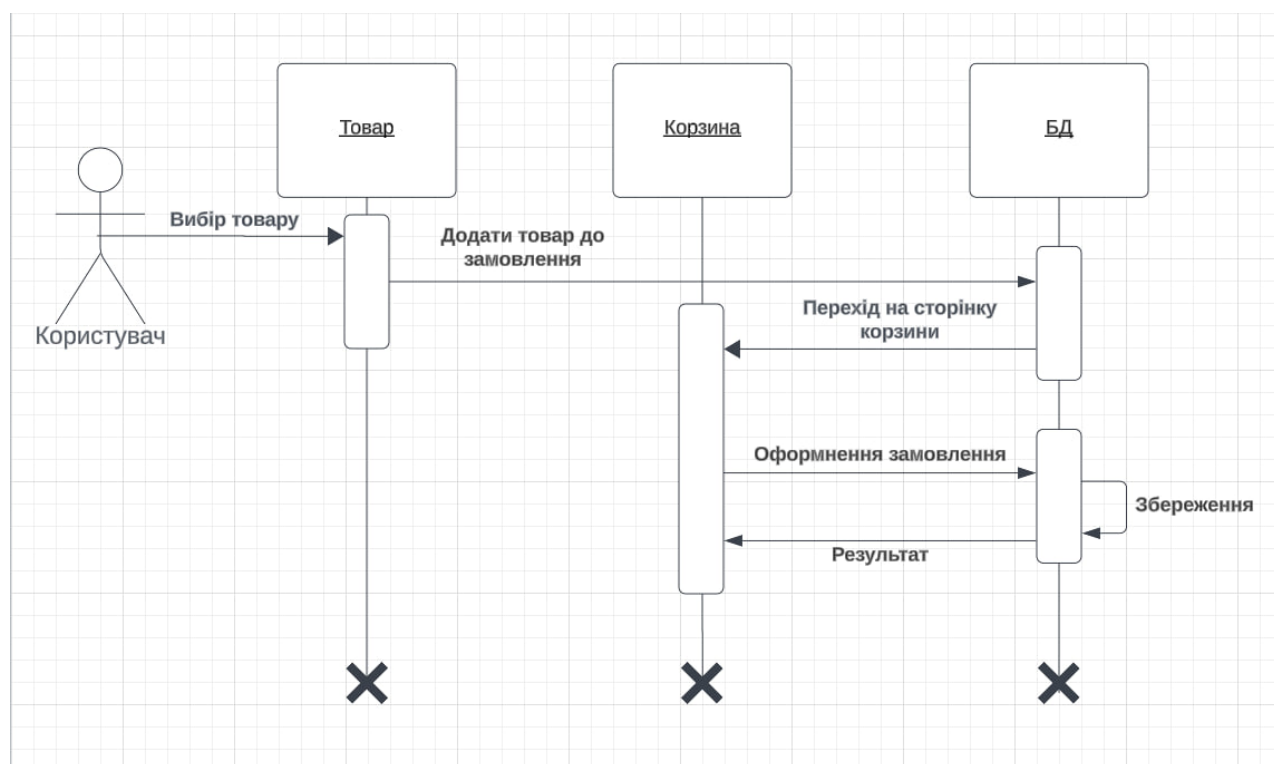


Рисунок 1.6 – Діаграма послідовності дій для здійснення замовлення

Як видно з рисунку 1.3.3 послідовні роботи користувача з інформаційною системою для здійснення замовлення наступна:

- користувач обирає товар. Попередньо він може здійснити пошук, фільтрацію товарів, що його потенційно цікавлять;
- обрані товари користувач додає до замовлення, формуючи кошик замовлення, що зберігається у базі даних системи;
- по закінченню формування кошика замовлення і його підтвердження, виконується оформлення замовлення, що також зберігається та опрацьовується адміністратором для виконання.

Отже, побудова загальної формалізованої моделі інформаційної системи виконана за допомогою UML діаграм поведінки та взаємодії, зокрема діаграм прецедентів та послідовності.

Структурні діаграми використаємо у наступних розділах при детальнішому проєктуванні, програмній реалізації інформаційної системи.

1.4 Висновки. Постановка задачі

Проведене у попередніх розділах дослідження, огляд та аналіз предметної області, вхідних та вихідних даних, існуючі рішення, аналогів, методів та засобів реалізації задачі, формалізоване моделювання інформаційної системи, дозволяють сформулювати постановку задачі та перейти до її реалізації.

Отже, у дипломному проєкті слід розробити інформаційну систему, що буде реалізована у вигляді веб додатку із можливістю пошук та підбору, фільтрації товарів медичного призначення. Використовуючи сучасні технології розробки веб сайтів слід розробити програмне забезпечення, яке забезпечить такі функціональні можливості:

- реєстрація та авторизація користувачів;

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

- додавання товарів;
- пошук, фільтрація товарів за обраними критеріям;
- додавання товарів в кошик;
- відправлення замовлення менеджеру;
- додавання сторінок для відображення інформації для користувача;
- передбачити адміністративну панель для можливості адміністратору наповнювати сайт товарами, створювати категорії товарів та інше;
- можливість створювати акційний товар;
- передбачити форму зворотного зв'язку для опрацювання звернень клієнтів.

Аналітичний огляд існуючих методів та засобів дозволяє здійснити обґрунтований вибір наступних технологій для реалізації завдання: мова програмування – php, фреймворк – laravel, СУБД – mysql, візуалізація – html та css.

Дослідження предметної області та формування постановки задачі дозволяють перейти до наступного етапу – проєктування інформаційної системи підбору товарів.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДБОРУ ТОВАРІВ

2.1 Проектування архітектури інформаційної системи підбору товарів

Проектування інформаційної системи потрібно розпочати з дослідження її архітектури.

Після аналізу предметної області та формування постановки задачі, визначено, що буде розроблятися інформаційна система у вигляді веб додатку, який передбачає використання клієнт-серверної технології (рисунок 2.1).

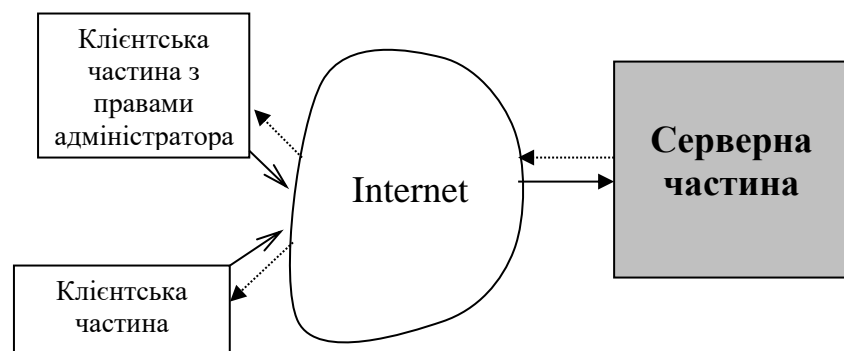


Рисунок 2.1 – Архітектура вебдодатку

З цього рисунку видно, що клієнтською частиною додатку буде програма-браузер, через яку користувачі будуть працювати із вебсайтом, що буде розташований на сервері. Сервер може бути виконаний як локальний сервер (емулятор), або ж реальний сервер (хостинг).

Клієнт-серверний додаток передбачає взаємодію з користувачем. Для реалізації взаємодії, спочатку сервер надішле користувачу відповідну html сторінку, на якій буде розташовуватись форма. Саме сюди юзер зможе вводити дані та відправляти їх на форму методом POST, натиснувши на кнопку на формі. Після цього до сервера надійдуть дані від користувача. Щоб отримати масив усіх

даних про додаток та його стан, на сервер приходять суперглобальні масиви, такі як `$_GET`, `$_POST`, `$_REQUEST`, `$_SESSION`. Щоб витягнути саме дані відправлені користувачем, нам потрібен масив `$_REQUEST`. У ньому у вигляді ключа-значення надходять значення.

Такий патерн передбачає розділення даних додатку і логіки на три компонента – модель, контролер та представлення, таким чином, що модифікація кожного компонента здійснюється незалежно (рисунок 2.2).

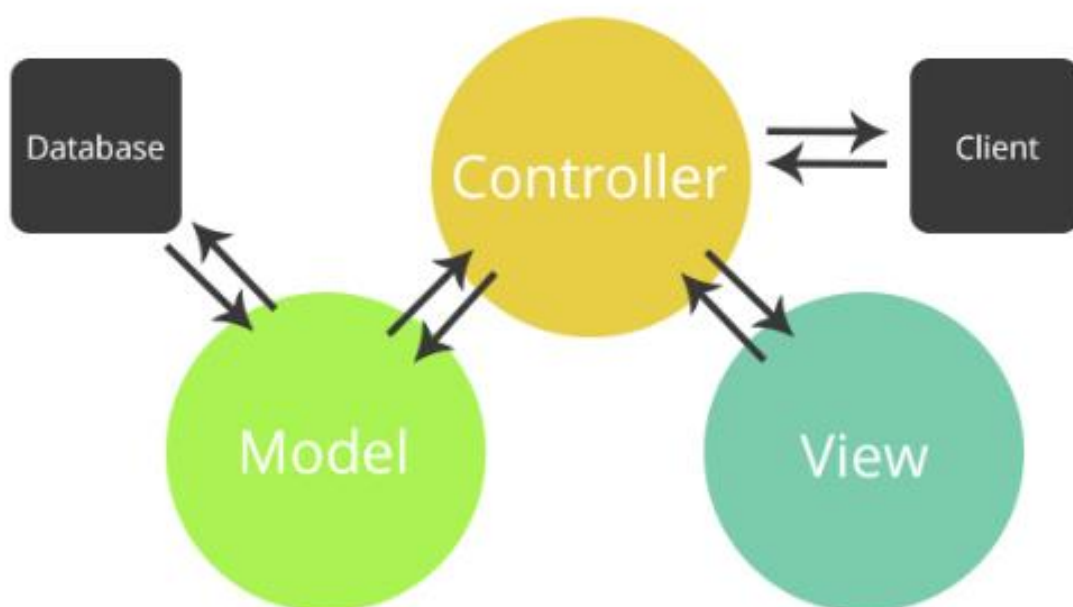


Рисунок 2.2 – Схема роботи MVC

Для проектування внутрішньої архітектури веб додатку слід зазначити, що буде використовуватись фреймворк Laravel.

Архітектура веб додатків, що реалізуються за допомогою такого фреймворку будується на основі архітектурного шаблону MVC.

Зм.	Арк.	№ докум.	Підпис	Дата

В архітектурі MVC кожен з компонентів має свою чітко визначену область відповідальності та допомагає розділити код Front-end та Back-end на окремі компоненти.

Модель зосереджена на логіці та обробці даних. Вона зберігає інформацію і визначає, як ця інформація має взаємодіяти з програмою.

Вигляд відповідає за відображення даних користувачеві. Це включає в себе графічний інтерфейс веб-сторінки або інші способи представлення інформації користувачу.

Контролер взаємодіє з користувачем та визначає, які дії виконувати на основі взаємодії користувача з Виглядом.

Він також відповідає за маршрутизацію і обробку запитів.

Архітектура проєктованого додатку буде використовувати можливості фреймворку Laravel, а тому й MVC, наступним чином.

У папці `routes/web.php` описані усі можливі маршрути нашого додатку та контролери, що будуть виконуватись, в разі переходу за певним шляхом.

Якщо там є відповідний запис, він буде вказувати, який саме контролер повинен виконатись.

Після цього буде виконуватись код в самому контролері. Шлях до контролера буде наступним: `app/Http/Controllers`. Вже безпосередньо в ньому ми можемо описати, які дані слід вибрати з бази та повернути користувачу та яку вивести сторінку (рисунок 2.3).

Також можливий такий розвиток подій, коли контролер не повертає дані у представлення, а робить пересилання на інший маршрут.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

Отже, `$_REQUEST` - це зручний спосіб отримання всіх даних, що передані у запиті, без врахування методу, яким вони були передані. Однак через ризики безпеки, пов'язані з `$_REQUEST`, рекомендується використовувати якнайбільш конкретні масиви (`$_GET` або `$_POST`), коли це можливо.

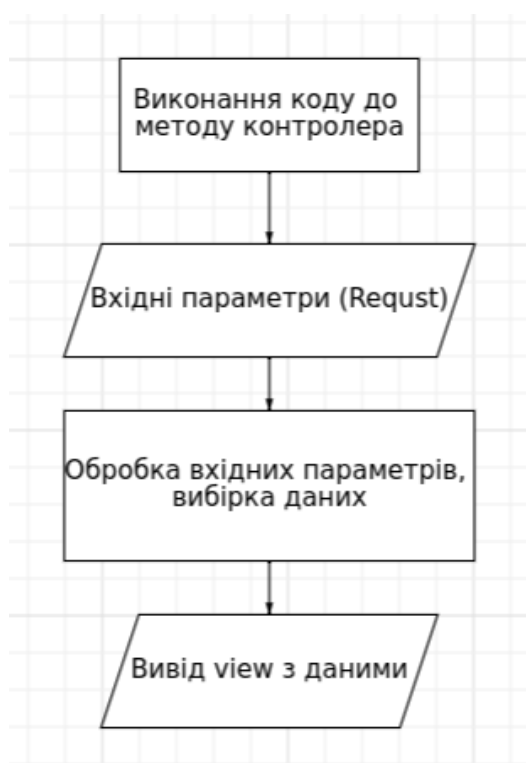


Рисунок 2.4 – Блок-схема роботи Арі контролера

Передача даних буде відбуватись лише у масивах. При отриманні даних від користувача або при виборі даних з бази даних буде присилатись масив з даними.

Враховуючи усе проведене вище дослідження особливостей роботи фреймворку Laravel, патерну MVC підсумкова архітектура інформаційної системи підбору товарів приведена на рисунку 2.5 у вигляді структурної UML діаграми компонентів.

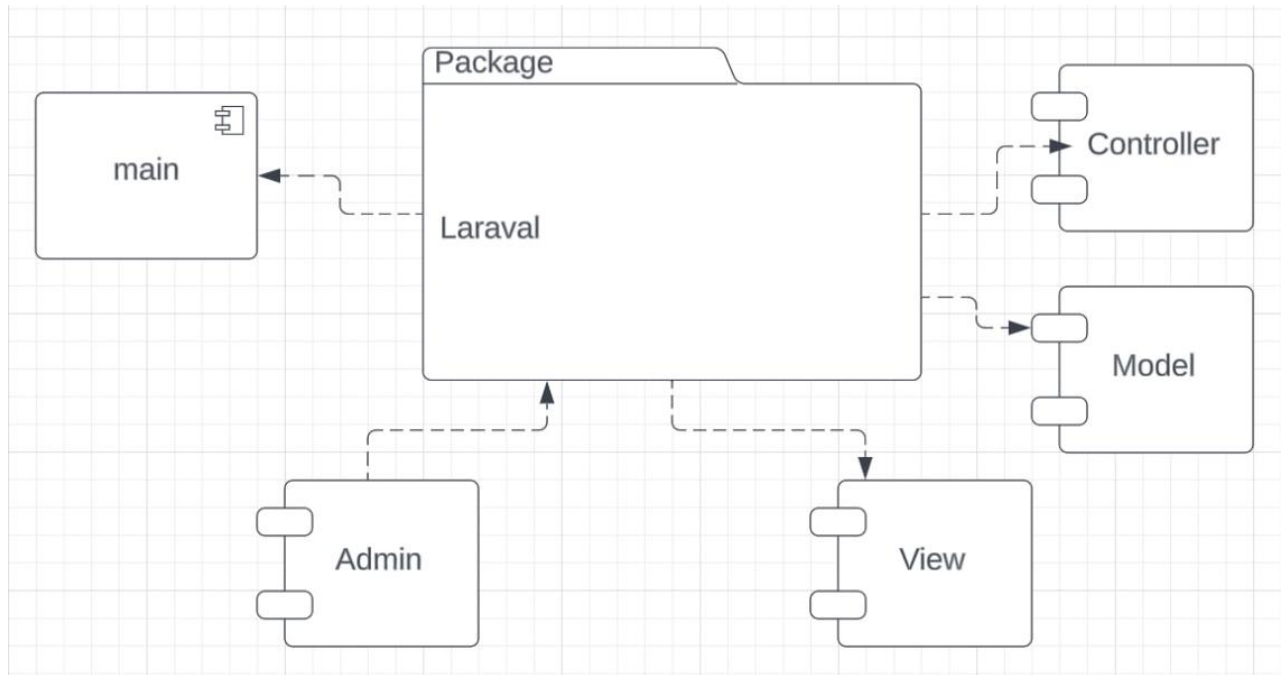


Рисунок 2.5 – Діаграма компонентів інформаційної системи підбору товарів

Перейдемо далі до проєктування окремих частин архітектури інформаційної системи.

2.2 Проєктування бази даних інформаційної системи підбору товарів

Однією із основних задач моделей, як частини архітектури MVC, є обробка даних. Для цієї задачі доцільно використати базу даних додатку, що буде основною структурою, що забезпечуватиме збереження та опрацювання інформації.

Спершу слід спроектувати базу даних. Процес проєктування реляційної бази даних полягає у її нормалізації.

Нормалізація бази даних – є процесом реорганізації даних, шляхом ліквідації груп, полів, що повторюються та інших протиріч, з метою приведення таблиць до вигляду, що дозволяє здійснювати не суперечливе та коректне

додавання даних, при цьому слід також забезпечити відсутність надлишковості даних.

У результаті проєктування бази даних має бути визначена її структура, склад таблиць, їхня структура та логічні зв'язки між ними. Таблиці визначаються складом стовпців, їхньою послідовністю, типом даних кожного стовпця та розміром, а також ключем таблиці.

Проєктування бази даних починається з вивчення предметної області. Далі потрібно визначити основні об'єкти предметної області та список усіх атрибутів, які повинні міститися в базі даних. Для цього слід звернутись до розділу «Аналіз вхідних та вихідних даних інформаційної системи підбору товарів», де така робота в основному була проведена.

Визначивши основну частину даних, розпочинаємо розробку структури бази. Далі розподіляємо поля генерального списку по базових таблицях. В першу чергу розподіл здійснюємо за функціональною ознакою, мета якого полягає у забезпеченні одноразового введення даних в одну таблицю по можливості в рамках одного підрозділу, потім здійснюємо подальшу нормалізацію даних (структуруємо усю вхідну інформацію, розкладаємо початкове відношення на кілька простих відношень меншого розміру) з метою вилучення повторів даних у таблицях бази даних.

Визначивши всі поля, ми отримаємо декілька сутностей, що містять усі необхідні атрибути предметної області. Ці сутності вже знаходяться в першій нормальній формі (1НФ), оскільки відповідають її вимогам. А саме вона передбачає, щоб кожне поле таблиці має бути неподільним і не має містити повторних груп. Це означає, що такі поля як ПІП та інші подібні поля повинні бути розділені на окремі поля, а поля типу «Вік» та «Дата народження», що по суті повторюються, не повинно бути.

Для другої нормальної форми потрібно, щоб база даних була приведена до першої нормальної форми, і дані у всіх не ключових стовпцях повністю залежали від первинного ключа і кожного елемента (стовпця) первинного ключа, якщо

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29

ключ є складеним. Під повною залежністю розуміються те, що значення в кожному не ключовому стовпці однозначно визначається значенням первинного ключа. Тому для кожної таблиці було визначено первинний ключ – відповідний код (ідентифікатор), який буде визначати унікальність кожного набору.

Також на цьому етапі були додані зовнішні ключі в таблиці для реалізації зв'язків між ними. Їх організують на основі спільного поля, в одній із таблиць воно обов'язково має бути ключовим.

Для третьої нормальної форми потрібно, щоб всі неключові стовпці таблиці не тільки залежали від первинного ключа таблиці, але були незалежними один від одного, тобто, щоб були відсутні транзитивні функціональні залежності між стовпцями таблиці. Для цього потрібно, щоб таблиці були попередньо приведені до першої та другої нормальної форми. При аналізі структури приведеної до другої нормальної форми, транзитивні функціональні залежності між стовпцями таблиць не виявлені, тому можна сказати, що структура знаходиться у 3-й нормальній формі.

Після виконання проведення нормалізації даних отримали БД для зберігання та оперування даними. Далі приведено детальний опис та результати проектування таблиць бази даних із зазначенням типу даних та призначення полів.

Певно, головною таблицею бази даних, буде таблиця «drugs», що буде містити усю необхідну інформацію про медичні товари, ліки (рисунок 2.6).

Призначення усіх інформативних полів цієї сутності було висвітлено у розділі «Аналіз вхідних та вихідних даних інформаційної системи підбору товарів», коли аналізувалась вхідна інформація про товари. Також були додані службові поля. А саме, Id – унікальний ідентифікатор, первинний ключ таблиці.

Враховуючи, що багато товарів можуть належати одній категорії товарів, було додано таблицю «categories», а для реалізації зв'язку «багато до одного» у таблицю «drugs» додали зовнішній ключ – поле categories_id.



Рисунок 2.6 – Таблиця «drugs»

Структура таблиці «categories» приведена на рисунку 2.7.



Рисунок 2.7 – Таблиця «categories»

Призначення таблиці «categories» та її полів зрозуміле. Зазначимо, що таблиця «categories» буде пов'язана зв'язком «один до багатьох» з таблицями «ats_categories» та «posts».

Для реалізації цих зв'язків використовується первинний ключ таблиці «categories», поле id та відповідні зовнішні ключі таблиць «ats_categories» та «posts», поля categories_id.

Структура таблиці «ats_categories» приведена на рисунку 2.8.



Рисунок 2.8 – Таблиця «ats_categories»

Таблиця «ats_categories» є класифікатором лікарських засобів за системою АТС. Структура таблиці «posts» приведена на рисунку 2.9.



Рисунок 2.9 – Таблиця «posts»

Таблиця «posts» потрібна для збереження та маніпулювання інформацією, що буде міститись у постах, повідомленнях про категорії товарів.

Повернемося до таблиці «drugs», що буде містити усю необхідну інформацію про медичні товари. Таблиця «drugs» буде пов'язана зв'язком «один до багатьох» з таблицями «hot_products», «analogs» та «drug_images».

Для реалізації такого зв'язку в ці таблиці слід додати зовнішній ключ, поле drugs_id. Це поле буде пов'язане з первинним ключем таблиці «drugs», полем id.

Структура таблиці «hot_products» показана на рисунку 2.10.

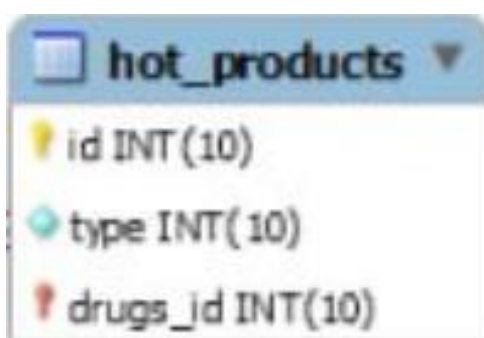


Рисунок 2.10 – Таблиця «hot_products»

Структура таблиці «analogs» показана на рисунку 2.11.

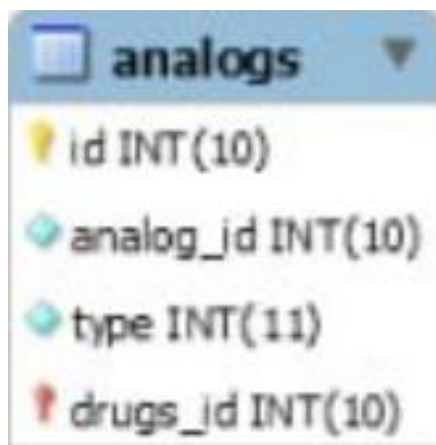


Рисунок 2.11 – Таблиця «analogs»

Структура таблиці «drug_images» показана на рисунку 2.12.

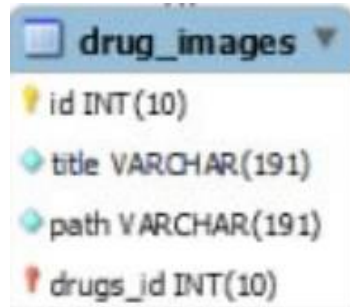


Рисунок 2.12 – Таблиця «drug_images»

Лікарські засоби, товари медичного призначення можуть відпускатись у різному вигляді: упаковках, пластинках, таблетках, сиробах і тому подібне. Тому слід створити ще одну таблицю «drug_portions», що містить інформацію не загальну інформацію про медичний товар, як в таблиці «drugs», а вже конкретний товар.

Структура цієї таблиці «drug_portions» показана на рисунку 2.13.

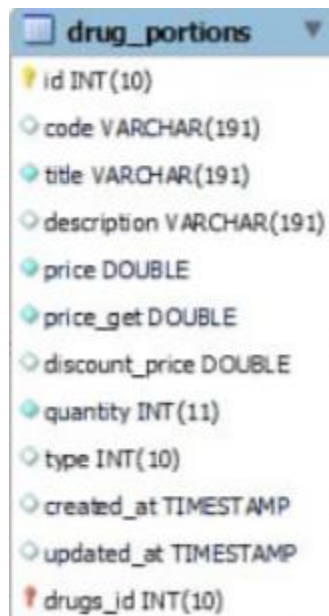


Рисунок 2.13 – Таблиця «drug_portions»

Призначення полів цієї таблиці також було висвітлено у розділі «Аналіз вхідних та вихідних даних інформаційної системи підбору товарів», коли

аналізувалась вхідна інформація про товари. Також були додані службові поля. А саме, Id – унікальний ідентифікатор, первинний ключ таблиці, поле drugs_id – зовнішній ключ. Це поле буде пов’язане з первинним ключем таблиці «drugs», полем id таким чином утворюючи зв’язок «багато до одного».

Ця таблиця «drug_portions» буде містити інформацію про конкретні товари, з якими й буде працювати кінцеві користувачі, для яких потрібно створити таблицю «users».

Структура цієї таблиці «users» показана на рисунку 2.14.

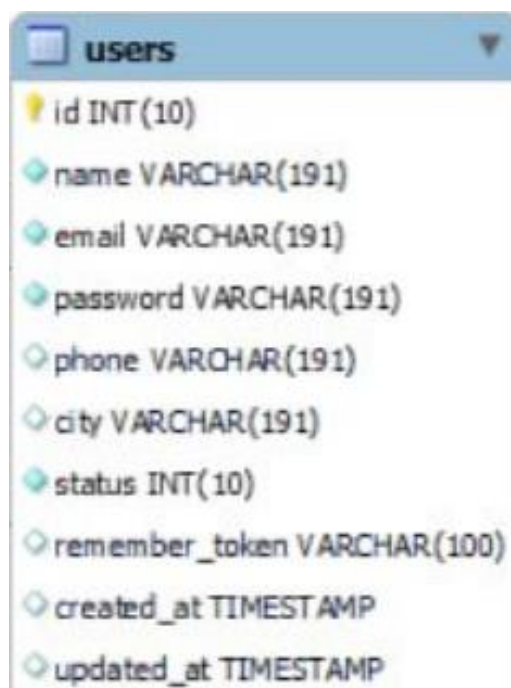


Рисунок 2.14 – Таблиця «users»

Призначення полів цієї таблиці також було висвітлено у розділі «Аналіз вхідних та вихідних даних інформаційної системи підбору товарів», коли аналізувалась вхідна інформація про користувачів. Додали поле id – унікальний ідентифікатор, первинний ключ таблиці.

Користувач інформаційної системи повинен мати можливість додавати товари у кошик. Тому слід створити відповідну таблицю.

Структура цієї таблиці «user_carts» (кошик) показана на рисунку 2.15.

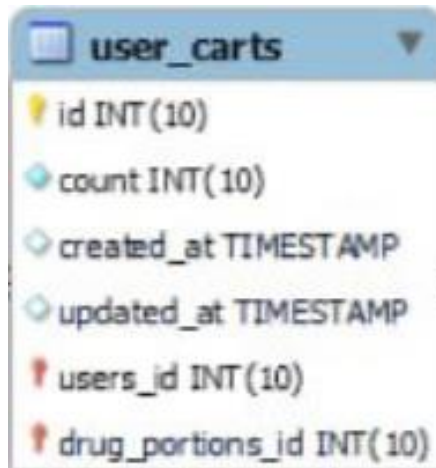


Рисунок 2.15 – Таблиця «user_carts»

Таблиця «user_carts» буде зв'язувати таблиці «users» та «drug_portions» відношенням «багато до багатьох». Для цього додали в її структуру зовнішні ключі users_id та drug_portions_id, які слід зв'язати з первинними ключами відповідних таблиць.

Схожа реалізація й задачі формування списку улюблених товарів. Для цього створюємо таблицю «favorites»

Структура цієї таблиці «favorites» (список улюблених товарів) показана на рисунку 2.16.

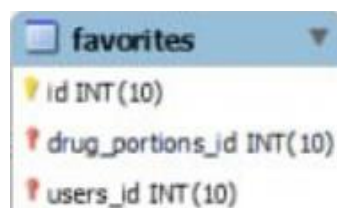


Рисунок 2.16 – Таблиця «favorites»

Вона також буде зв'язувати таблиці «users» та «drug_portions» відношенням «багато до багатьох». Для цього додали в її структуру зовнішні

ключі `users_id` та `drug_portions_id`, які слід зв'язати з первинними ключами відповідних таблиць.

Також користувач інформаційної системи повинен мати можливість сформулювати та здійснити замовлення товарів. Для формування замовлення слід створити відповідну таблицю «orders».

Структура таблиці «orders» (замовлення) показана на рисунку 2.17.

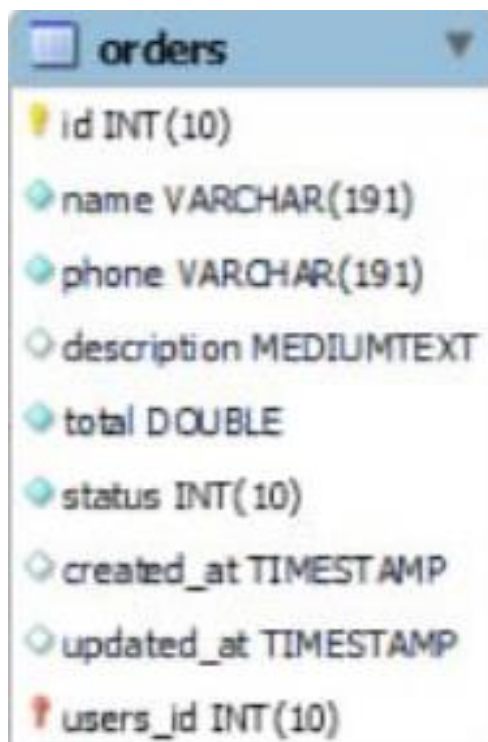


Рисунок 2.17 – Таблиця «orders»

Оскільки користувачі можуть створити багато замовлень, відповідні таблиці «users» та «orders» зв'яжемо відношенням «один до багатьох». Для цього додаємо у таблицю «orders» зовнішній ключ `users_id` та зв'язуємо його із первинним ключем `id` таблиці «users».

Щоб наповнити конкретне замовлення товарами, слід створити ще одну таблицю «order_items», структура якої зображена на рисунку 2.18.

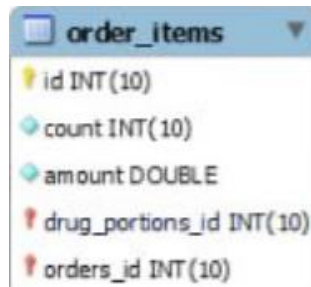


Рисунок 2.18 – Таблиця «order_items»

Призначення полів цієї таблиці, як і таблиці «orders» було висвітлено у розділі «Аналіз вхідних та вихідних даних інформаційної системи підбору товарів», коли аналізувалась вхідна інформація про замовлення товарів.

Ця таблиця «order_items» буде зв'язувати таблиці «drug_portions» та «orders» відношенням «багато до багатьох». Для цього додали в її структуру зовнішні ключі orders_id та drug_portions_id, які слід зв'язати з первинними ключами відповідних таблиць.

На цьому нормалізацію таблиць бази даних, формалізацію зв'язків закінчено, інфологічна схема бази даних приведена на рисунку 2.19.

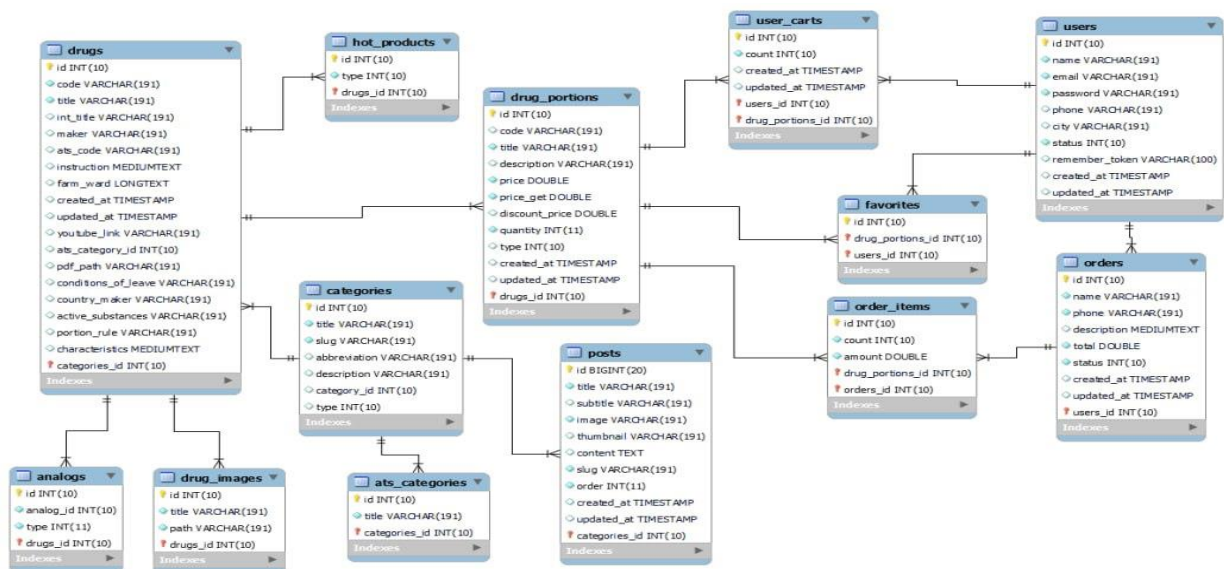


Рисунок 2.19 – Схема бази даних інформаційної системи підбору товарів

На цьому проєктування бази даних, як частини архітектурної компоненти Модель (Model) патерну MVC завершено, перейдемо до дизайну інтерфейсу інформаційної системи.

2.3 Проєктування інтерфейсу інформаційної системи підбору товарів

Згідно архітектурного шаблону MVC, за взаємодію з користувачем, тобто за інтерфейсну частину інформаційної системи відповідає компонент Вид (View). Тому, власне, завершальним етапом реалізації архітектурного шаблону MVC й буде проєктування інтерфейсу інформаційної системи підбору товарів.

Для проєктування інтерфейсу веб сторінок додатку створимо основний загальний макет, шаблон інтерфесу сайту (рисунок 2.20).

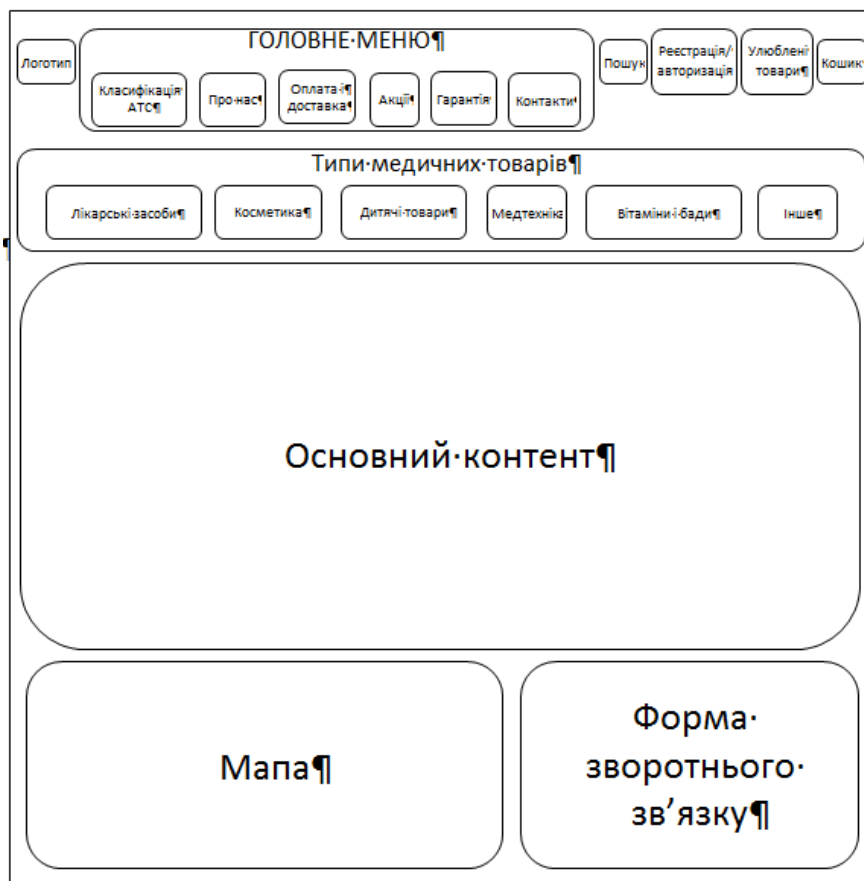


Рисунок 2.20 – Головний шаблон інтерфейсу сайту

Цей шаблон будемо використовувати для реалізації інтерфейсу усіх сторінок веб додатку. Фактично на таких сторінках буде лише відрізнятись контент, який буде виводитись у відповідному полі. Верхня й нижня ж частина сторінки залишиться незмінною.

Так на головній сторінці у полі контенту будуть виводитись найактуальніші медичні товари (актуальний вибір), а нижче акційні товари.

При виборі пункту головного меню «Класифікація АТС» відкриється відповідна сторінка сайту, де в полі контенту буде виведено, власне, таку класифікацію (рисунок 2.21).

Головна / Класифікація АТС

Класифікація АТС



- А ЗАСОБИ, ЩО ВПЛИВАЮТЬ НА ТРАВНУ СИСТЕМУ І МЕТАБОЛІЗМ



- А01 ЗАСОБИ ДЛЯ ЗАСТОСУВАННЯ В СТОМАТОЛОПІ



- В ЗАСОБИ, ЩО ВПЛИВАЮТЬ НА СИСТЕМУ КРОВІ ТА ГЕМОПОЕЗ



- С ЗАСОБИ, ЩО ВПЛИВАЮТЬ НА СЕРЦЕВО-СУДИННУ СИСТЕМУ



- D ДЕРМАТОЛОПІЧНІ ЗАСОБИ



- G ЗАСОБИ, ЩО ВПЛИВАЮТЬ НА СЕЧОСТАТЕВУ СИСТЕМУ ТА СТАТЕВІ ГОРМОНИ



- H ПРЕПАРАТИ ГОРМОНІВ ДЛЯ СИСТЕМНОГО ЗАСТОСУВАННЯ (ОКРІМ СТАТЕВИХ ГОРМОНІВ ТА ІНСУЛІНІВ)



- J ПРОТИМІКРОБНІ ЗАСОБИ ДЛЯ СИСТЕМНОГО ЗАСТОСУВАННЯ



- L АНТИНЕОПЛАСТИЧНІ ТА ІМУНОМОДУЛЮЮЧІ ЗАСОБИ



- M ЗАСОБИ, ЩО ВПЛИВАЮТЬ НА ОПОРНО-РУХОВИЙ АПАРАТ



- N ЗАСОБИ, ЩО ДІЮТЬ НА НЕРВОВУ СИСТЕМУ

Рисунок 2.21 – Класифікація АТС

Зм.	Арк.	№ докум.	Підпис	Дата

КВРІСТ.101041. 21.01.03 ПЗ

Арк.

40

Наступний пункт меню «Про нас» веде до сторінки з відповідною інформацією контенту та переліком сертифікатів.

Так само й пункт меню «Доставка і оплата» веде до сторінки з інформацією про способи доставки, оплати замовлень

Пункт меню «Акції» відкриє сторінку, що містить акційні пропозиції товарів. Характерно, що в цій сторінці слід передбачити бокову панель для підбору потрібного товару (рисунок 2.22).

Всі товари

Лікарські засоби ▾

Косметика ▾

Дитячі товари ▾

Медтехніка ▾

Вітаміни і БАДи ▾

Ціна ▲

0 - 137

Форма випуску ▲

Упаковка

Пластина

Капсули

Коробка

Країна виробник ▲

Німеччина

Англія

Іспанія

Умови відпуску ▲

За рецептом

Без рецепту

Рисунок 2.22 – Панель для підбору потрібного товару

Також додамо в контент поле зі списком сортування товарів.

Зазначимо, що при виборі будь якого типу медичних товарів з панелі «Типи медичних товарів», що знаходиться нижче головного меню додатку, будуть виводитись товари відповідного призначення, типу, а також аналогічна бокова панель для підбору потрібного товару і поле сортування.

Наступний пункт меню «Гарантія» веде до сторінки з переліком сертифікатів, що забезпечують та гарантують діяльність такого типу.

Щоб переглянути контактні дані, слід обрати пункт меню «Контакти» й перейти на відповідну сторінку.

Для перегляду ширшої інформації про товар, достатньо натиснути лівою кнопкою миші на зображенні потрібного товару.

Ця інформація буде виводитись на сторінці, що матиме вигляд, що показано на рисунку 2.23.

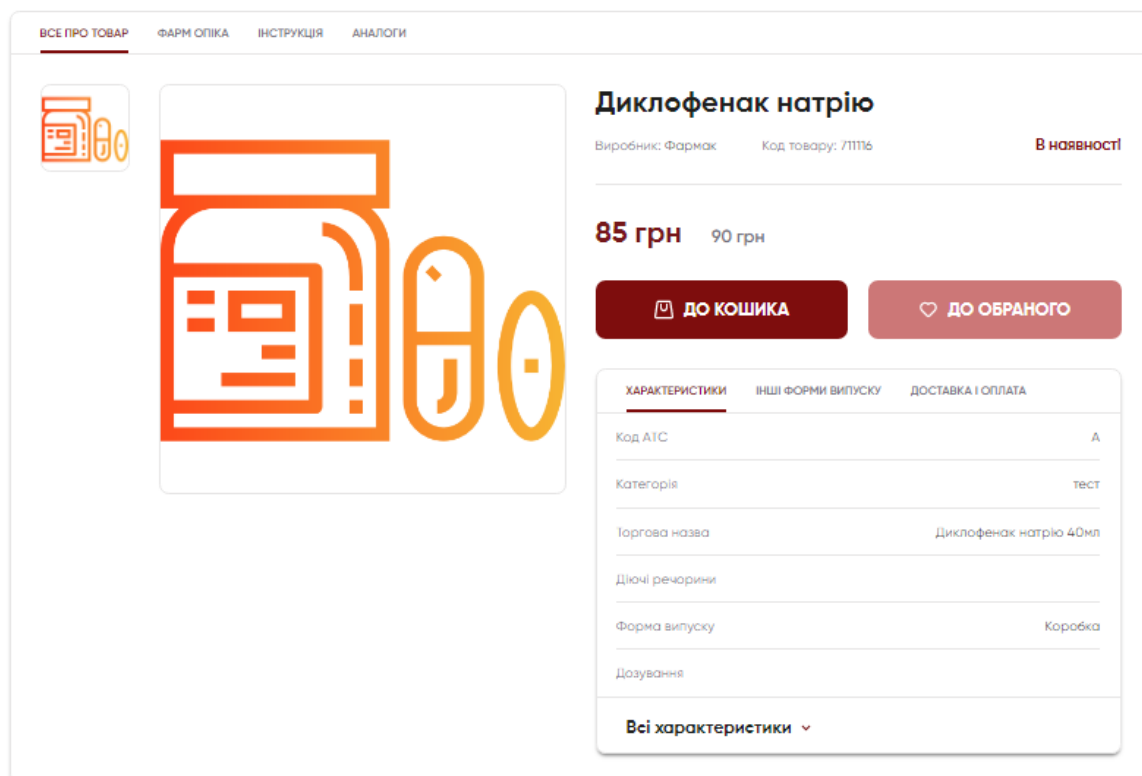




Рисунок 2.23 – Інформація про товар

Зм.	Арк.	№ докум.	Підпис	Дата

Як видно з рисунка 2.23 користувач може переглянути зображення товару, текстові характеристики, опис та додати за бажанням його в кошик чи до обраного. після цього можна продовжувати підбір товарів, або ж перейти в кошик для здійснення замовлення, купівлі. Сторінка «Кошик» має наступний інтерфейс (рисунок 2.24).

Корзина

	Нутрікомп Стандарт Нейтральний у пластиковій пляшці 500 мл. Нутрікомп Стандарт Нейтральний у пластиковій пляшці 500 мл	- 2 +	121.74 грн	⊗
	Захисний крем Молікар Скін, 200мл. Захисний крем Молікар Скін, 200мл	- 2 +	196.84 грн	⊗

Загальна сума 637.16 грн
Очистити корзину

Іван Тхір *

+380(98)989-8989 *

Додаткові побажання

КУПИТИ

Рисунок 2.24 – Інтерфейс сторінки «Кошик»

Зм.	Арк.	№ докум.	Підпис	Дата

КВРІСТ.101041. 21.01.03 ПЗ

Арк.

43

Для забезпечення реєстрації чи авторизації користувачів спроектуємо та реалізуємо форми, інтерфейс яких показано на рисунку 2.25.

Рисунок 2.25 – Інтерфейс форм для реєстрації чи авторизації користувачів

Для адміністрування веб додатку слід спроектувати інтерфейс адміністративної панелі, у якій вони зможуть переглядати, редагувати, видаляти та додавати дані різного типу (рисунок 2.26).

	<Меню>	
Заголовок		
Інформація 1	X	редагув.
Інформація 2	X	редагув.
Інформація 3	X	редагув.
Інформація 4	X	редагув.

Рисунок 2.26– Макет сторінки адміністратора для редагування даних

На цій панелі навігація буде розміщена з лівої частини екрана, а справа буде виводитись контент відповідного, вибраного в навігаційній панелі пункту (рисунок 2.27).

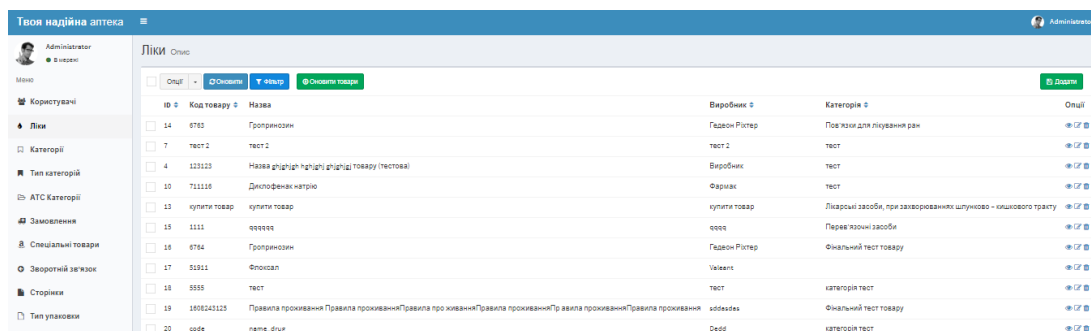


Рисунок 2.27 – Інтерфейс адміністративної панелі

На цьому проектуванні інтерфейсу інформаційної системи підбору товарів буде завершено.

2.4 Висновки

Отже, у цьому розділі було проведено проектування інформаційної системи підбору товарів. Було визначено основне архітектурне рішення створення програмного продукту, представлено та обґрунтовано використання архітектурного шаблону. Також було проведено проектування багатопланової архітектури на основі MVC, описано основні компоненти програмної системи, які взаємодіють при розробці та використанні користувачами.

На основі вхідних та вихідних даних було спроектовано та нормалізовано структуру реляційної бази даних, що була оформлена у вигляді інфологічної схеми.

Далі було спроектовано користувацький інтерфейс та інтерфейс адміністратора інформаційної системи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДБОРУ ТОВАРІВ

3.1 Програмна реалізація веб додатку

Програмну реалізацію задачі розпочнемо із налаштування програмного проєкту.

Для підключення і управління сторонніми бібліотеками або пакетами в PHP-проєкті необхідно встановити Composer. Для його встановлення потрібно перейти на офіційний сайт завантажити застосунок відповідно до своєї системи. В нашому випадку потрібно завантажити файл `Composer-Setup.exe`.



Рисунок 3.1 – Вигляд фрагменту офіційного сайту <https://getcomposer.org/> для завантаження файлу застосунку

Робота з Composer у XAMPP за умовчанням здійснюється у системній консолі. Для того щоб відкрити консоль необхідно натиснути комбінацію клавіш “Win + R”, після чого з’явиться вікно для введення команди у якому далі потрібно ввести команду «cmd» та натиснути на кнопку «ОК».

У консолі для перевірки того, що Composer підключено, наприклад, можна ввести команду: `composer`.

Ця команда відобразить версію Composer (рисунок 3.2).

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\ruhi>composer

Composer

Composer version 1.3.0 2016-12-24 00:47:03

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display this help message
  -q, --quiet           Do not output any message
  -V, --version         Display this application version
  --ansi                Force ANSI output
  --no-ansi            Disable ANSI output
```

Рисунок 3.2 – Перевірка працездатності Composer

Якщо під час виконання цієї команди з'явиться повідомлення, що версія Composer застаріла, її можна оновити. Здійснюється це шляхом виконання наступної команди: `composer self-update`.

Після встановлення «composer» необхідно встановити Laravel. Для встановлення Laravel та створення проєкту необхідно ввести такі команди в консолі:

```
composer create-project laravel/laravel domains/farmacy.lrvl
```

Після введення вище наведеної команди у нашій консолі відразу почнеться створення нашого проєкту Laravel, за чим ми зможе спостерігати у данному консольному вікні. Створення проєкту займає приблизно від декількох секунд, до декількох хвилин (рисунок 3.3).

```
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
71 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
No security vulnerability advisories found.
> @php artisan key:generate --ansi
Application key set successfully.
```

Рисунок 3.3– Створення проєкту Laravel

Для початку роботи з проєктом та редагування програмного коду слід завантажити IDEA PHPStorm, що є безкоштовною для студентів.

Відкриємо створений проєкт у PHPStorm. Після цього отримаємо наступну файлову структуру (рисунок 3.4).

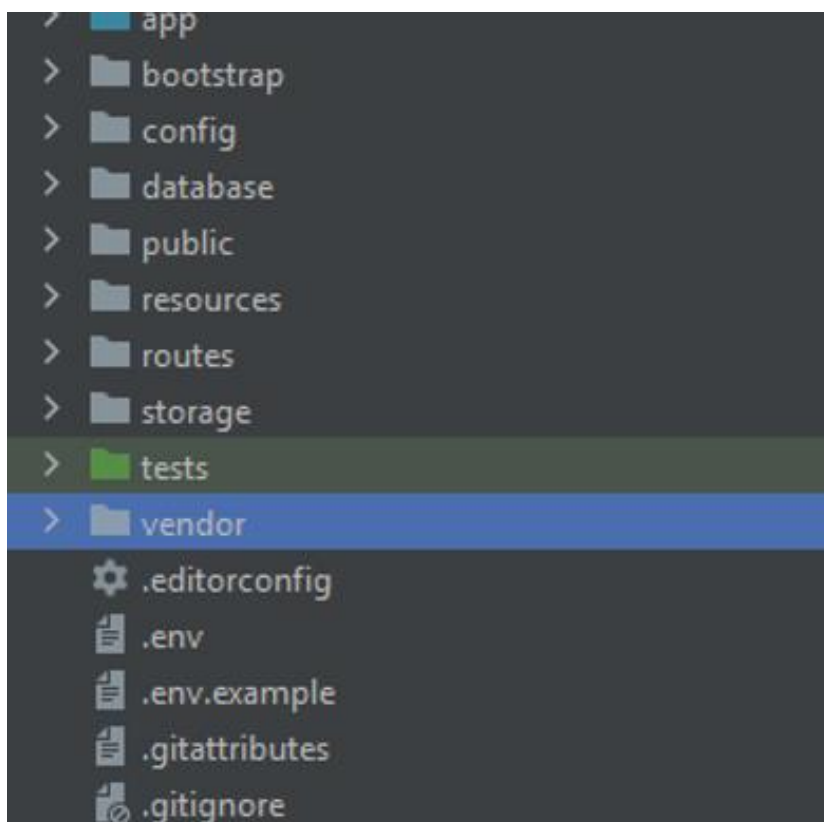


Рисунок 3.4 – Файлова структура фреймворку Laravel

Як бачимо, зліва знаходиться перелік баз даних, що містяться на сервері. З правого боку вікна розташована робоча область, вгорі якої є меню, за допомогою якого з легкістю виконуються визначені команди.

Натиснувши на посилання з назвою «New», що розташована над переліком баз даних, відкриється у робочій області вигляд для створення нової бази даних та запропонує ввести її назву. Введемо назву «pharmacy_db» та натиснемо на кнопку «Створити», після чого відбудеться процес створення БД і у разі успішного виконання з'явиться відповідне повідомлення (рисунок 3.6).

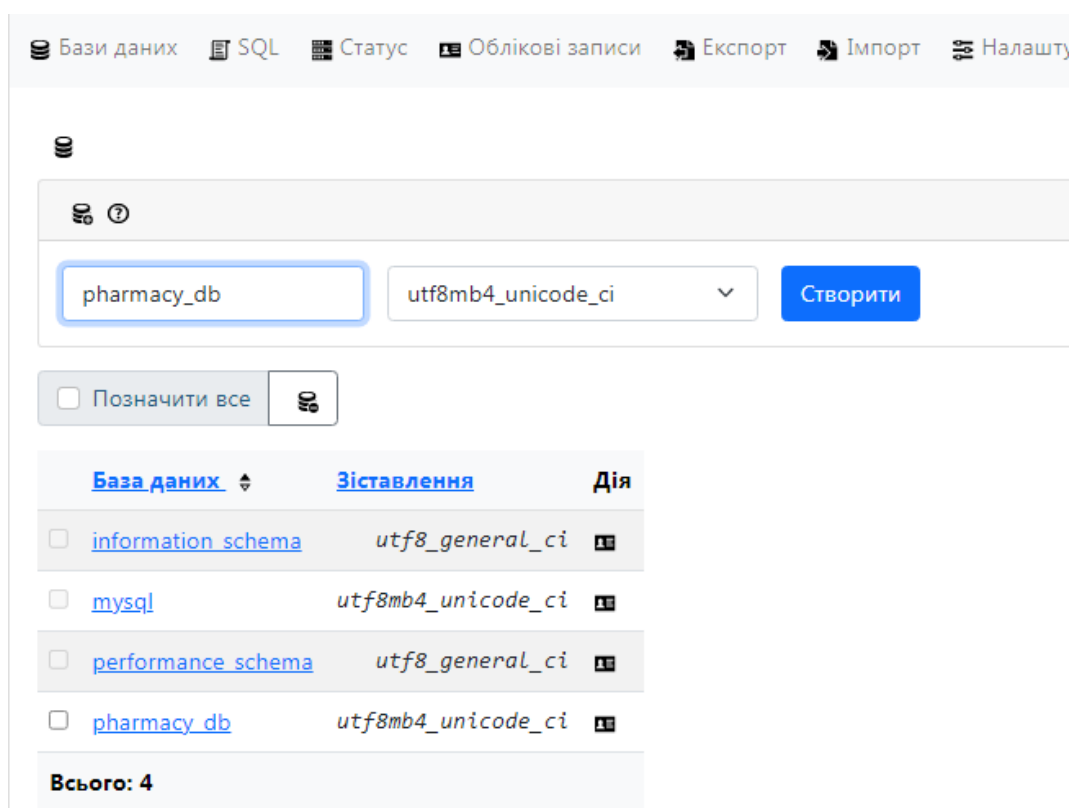


Рисунок 3.6 – Створення бази даних

Тепер коли ми створили власну базу, потрібно підключити її до проєкту. Для цього у корені проєкту у файлі .env потрібно вказати параметри підключення до бази даних (рисунок 3.7).

```
APP_ENV=local
APP_DEBUG=true
APP_KEY=base64:FUUzqezbPUoY6H5ggy9opkKM6XRSPsFKXu+HJQUIRLg=
APP_URL=http://Farmacy.lrv1

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=pharmacy_db
DB_USERNAME=root
DB_PASSWORD=

CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587 #465
MAIL_USERNAME=testmail@gmail.com
MAIL_PASSWORD=setpassword
MAIL_ENCRYPTION=tls #ssl
```

Рисунок 3.7 – Підключення бази даних до проєкту

Спроектовану інфологічну модель потрібно запрограмувати. Для цього слід створити таблиці бази даних відповідно до інфологічної моделі, полям таблиць задати відповідні назви та надати необхідні типи даних. Також при необхідності можна задати ключовим полям значення «Auto Increment», для того щоб уникнути дублювання полів та створити зв'язки між таблицями бази даних відповідно до спроектованої схеми БД. Але створення таблиць буде виконуватись не з використанням можливостей оболонки phpMyAdmin, а використовуючи механізм міграцій.

У фреймворку Laravel таблиці створюються наступним чином: створюється нова міграція, яка створює новий файл в папці бази даних, в якому потрібно вказати полі таблиці, щоб створити таблицю та що потрібно робити при відміні міграції.

Отримавши підключення до бази даних, далі потрібно налаштувати міграції для створення вмісту БД.

Міграції – це механізм, що дозволяє вміщувати в собі зміни бази даних. Чим зручні міграції:

- міграції можна застосовувати і відкочувати;
- за допомогою цього механізму базу даних проекту можна дуже швидко привести в актуальний стан на будь-якому комп'ютері. Це зручно як при командній розробці, так і при оновленні сайту на сервері;
- не треба зберігати безліч sql файлів для відстеження змін. Механізми міграцій беруть цю задачу на себе

Основні операції, які повинна реалізовувати кожна міграція:

- up – розгорнути зміни поточної міграції;
- down – відкочується зміни міграції.

Отже міграції це щось на зразок системи контролю версій для бази даних.

Для створення нової міграції потрібно виконати в командному рядку *artisan*-команду *make:migration*:

```
php artisan make:migration
```

В результаті чого створиться файл у папці *database/migrations*, в контенті якого потрібно прописати поля таблиці бази даних відповідно до спроектованої схеми БД.

Далі потрібно в міграції відповідно до розробленої схеми бази даних, прописати структуру кожної таблиці.

Аналогічно створюємо усі наступні міграції. Після цього запускаємо міграцію командою:

```
php artisan migrate
```

В результаті буде створено усі таблиці в базі даних.

Тепер для роботи із даними потрібно створити моделі, які відображатимуть структуру сутностей та відношення із іншими моделями, реалізуючи зв'язки між таблицями. Для зручності доступу до колекцій даних при запитах до бази даних будемо використовувати ORM, що вбудована у фреймворк Eloquent. Він значно спростить нам процес витягування даних із бази, що дасть нам змогу максимально зручно і швидко опрацьовувати данні.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

Наведемо приклад створення зв'язку із іншою моделлю. Тобто вкажемо, що поточна сутність може містити колекцію даних іншої зазначеної моделі. Таким чином організували зв'язок між таблицями один до багатьох.

```
public function portions()  
{  
    return $this->hasMany(DrugPortion::class);  
}
```

Для зворотнього зв'язку потрібно прописати наступний код у відповідній сутності:

```
public function drug()  
{  
    return $this->belongsTo(Drug::class);  
}
```

Для того, щоб зручніше було працювати із сутностями створимо репозиторії, в яких будуть міститися методи, що стосуватимуться безпосередньо конкретних таблиць.

Далі перейдемо до створення контролерів. Вони мають містити методи, які будуть опрацьовувати запити від користувача.

Отже, створимо контролер для товарів:

```
php artisan make:controller DrugController
```

В контролері опишемо методи, які мають опрацьовувати дані та видати їх на екран користувачу.

Для того, щоб використати метод контролера потрібно, щоб він опрацьовував певний запит від користувача, тобто якусь електронну адресу на сайті. Для цього потрібно в проєкті в папці routes в файлі web.php зареєструвати роут і власне який метод конкретного контролера опрацьовуватиме той запит.

Також в контролері прописується логіка пошуку товару за обраними критеріями.

З контролера дані які опрацьовувались будуть виведені в файл розмітки, що знаходиться в папці ресурсів resources/pages в файл product_card.blade.php (рисунок 3.8).

Власне в цьому файлі знаходиться основний каркас сторінки, а також підключення файлів стилів та скриптів.

Файли стилів та скриптів знаходяться у наступній папці (рисунок 3.9).

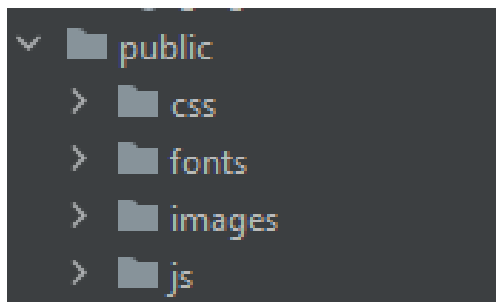


Рисунок 3.9 – Файли стилів та скриптів

Оскільки в системі мають бути, як звичайні користувачі, що шукатимуть та замовлятимуть товар, так і адміністратор, який має мати можливість наповнювати сайт товарами та приймати замовлення, слід створити адміністративну панель.

Для створення адміністративної панелі скористаємося паунком `encore/laravel-admin`. Для цього слід його встановити через `composer`:

```
composer require encore/laravel-admin
```

Потім слід виконати команду для розгортання:

```
php artisan vendor:publish
```

А також для встановлення додаткових модулів та розгорнення міграції паунку потрібно виконати команду:

```
php artisan admin:install
```

Після цього в системі створиться папка для створення контролерів суто для адміністративної частини.

В контролерах для адміністративної частини потрібно зазначати елементи які потрібні, щоб були на сторінках (сконфігурувати).

Відповідно до структури Laravel для роботи із контролером його потрібно зареєструвати в файлі роутів, що згенерувався після встановлення паунку.

```
$router->resource('/drugs', 'DrugController');
```

З використанням цих роутів в адміністративній панелі створюється відповідне меню, за допомогою якого можна з легкістю переміщатися між сутностями.

3.2 Тестування проєкту

Тестування програмного забезпечення – перевірка відповідності між реальною і очікуваною поведінкою програми, що здійснюється на кінцевому наборі тестів, обраному певним чином.

Цілі тестування:

- підвищити ймовірність того, що додаток, призначений для тестування, буде працювати правильно при будь-яких обставинах та переданих йому даних;
- підвищити ймовірність того, що додаток, призначений для тестування, буде відповідати всім описаним вимогам;
- надання актуальної інформації про стан продукту на момент здійснення замовлення.

Існує велика кількість різноманітного тестування: модульне, функціональне, на безпеку, на продуктивність, GUI тестування та інші.

3.2.1. Функціональне тестування

Під час фази розроблення тестування було спроектовано 6 тестових випадків для функціонального тестування. Кожен тестовий випадок містить детальні кроки для виконання, тестові дані, очікувані результати та відповідні результати, які були одержані при виконанні дії, тестуючи програмне забезпечення. Результати тестування приведені у таблиці 3.1.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

Таблиця 3.1 – Тестові випадки функціонального тестування

№	Steps (Кроки/дії)	Expected Results (Очікувані результати)	Actual results (Реальні результати):
1.	Запуск програми	Після переходу на сайт, завантажуються головна сторінка.	Завантажилась головна сторінка
2.	Залишити коментар, відгук.	Після введення даних у формі відгуку та натискання кнопки «надіслати», має сформуватись відгук.	Інформація відгуку з'явилась.
3.	Ввести критерії пошуку медичного товару в рядок пошуку	Після введення критеріїв пошуку мають відобразитись відповідні результати.	Відобразились пропоновані результати пошуку.
4.	Завершення роботи	Після натискання кнопки «Вихід» інформація про користувача очищується з сесії.	Інформація про користувача очистилась.
5	Ввід не коректної e-mail адреси в підписці.	Зупинка роботи програми, вивід повідомлення про помилку.	Програма не почала обробку даних, вивелось повідомлення.
6.	Додавання товарів у корзину і подальше створення замовлення.	Формування замовлення та вивід інформації про це в адміністративній частині додатку.	В частині «Замовлення» з'явилося нове замовлення.

Зм.	Арк.	№ докум.	Підпис	Дата

6 тестів із 6 наведених в таблиці 3.1 «Тестові випадки функціонального тестування» пройшли успішно, отже, функціональне тестування розглядається як успішне.

3.2.2. GUI тестування

Метою тестування користувацького інтерфейсу є перевірка правильності навігації по об'єкту тестування та відповідність назв кнопок їх діям. Для цього усі елементи керування та навігації будуть перевірені вручну. Було створено бтестових випадків, результати яких відображенні у таблиці 3.2.

Таблиця 3.2. – Випадки тестування користувацького інтерфейсу

№	Steps (Кроки/дії)	Expected Results (Очікувані результати)	Actual results (Реальні результати):
1.	Натиснути на кнопку «Вхід у систему»	Авторизація користувача, або вивід повідомлення про помилку	Авторизація користувача
2.	Натиснути на кнопку «Створити обліковий запис»	Відкриття форми для реєстрації нового користувача.	Завантажилась форма реєстрації, відбулась реєстрація користувача
3.	Натиснути на пункті меню «Акції»	Поява сторінки із акційними товарами.	Відкрилась сторінка із акційними товарами.
4.	Натиснути на кнопку «Замовити зворотній дзвінок»	Відкриття форми для замовлення зворотного дзвінка.	Завантажилась форма замовлення зворотного дзвінка.

4 тести із 4 наведених в таблиці 3.2.2. «Випадки тестування користувацького інтерфейсу» пройшли успішно. Отже, тестування користувацького інтерфейсу розглядається як успішне.

3.2.3. Модульне тестування

Фреймворк Laravel надає різноманітні можливості для тестування веб додатків, що створюються із його використанням. Одним із найпоширеніших видів тестування є модульне тестування, що також можна реалізувати використовуючи можливості Laravel.

У Laravel тестування використовується для перевірки роботи вашого додатку на належність специфікаціям і очікувань. Для написання тестів в Laravel використовуються фреймворки PHPUnit і Laravel Dusk (для тестування веб-інтерфейсу).

Модульне тестування в Laravel - це процес тестування окремих компонентів вашого додатку (наприклад, контролерів, моделей, сервісів) для перевірки їх правильності та відповідності специфікаціям. Laravel надає різні можливості для модульного тестування, і однією з найпоширеніших бібліотек для цього є PHPUnit.

Для цього можна запустити стандартні вбудовані тести. Для цього слід виконати наступні команди (рисунок 3.10).

```
php vendor/phpunit/phpunit/phpunit
```

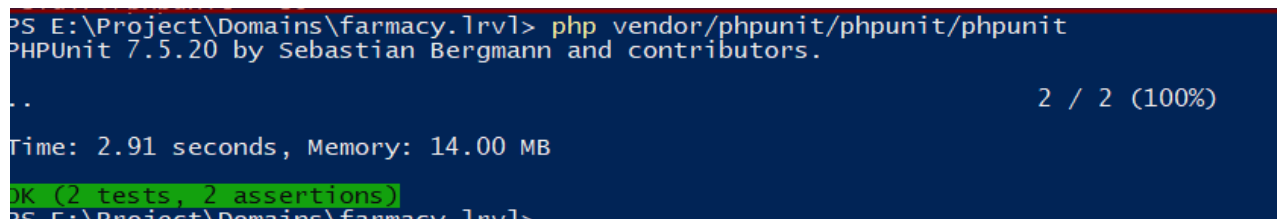


Рисунок 3.10 – Виконання модульних тестів Laravel

Щоб створити власні тести слід виконати наступну команди:

```
php artisan make:test drugTest
```

Отже, створимо тест (рисунок 3.11).

```
php artisan make:test drugTest -unit
```

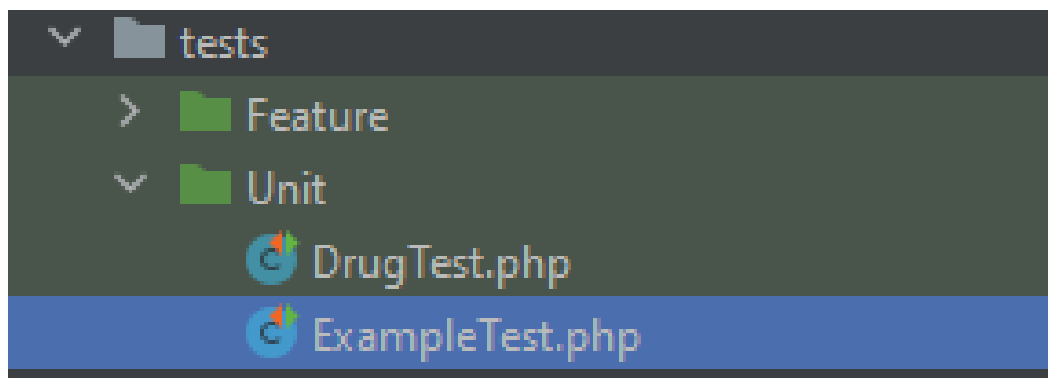


Рисунок 3.11 – Створення власного модульного тесту drugTest

Бачимо, що наш користувацький тест drugTest.php успішно створено, і має такий початковий вигляд (рисунок 3.12).

```
class DrugTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testExample()
    {
        $this->assertTrue( condition: true);
    }
}
```

Рисунок 3.12 – Початковий вигляд тесту drugTest.php

Результати виконання тесту drugTest успішні й приведені на рисунку 3.13.

3.3 Інструкція з інсталяції та експлуатації

Для забезпечення інсталяції та експлуатації розробленого програмного продукту потрібна наявність комп'ютера із необхідним програмним та апаратним забезпеченням.

До програмного забезпечення в першу чергу відноситься операційна система. Даний програмний продукт розроблявся як веб-сайт, тому для його використання не важлива операційна система, проте використаний для розробки веб-сервер працює лише з сімейством операційних систем Windows. Для роботи сайту із базою даних необхідна наявність системи керування базами даних MySQL, але її встановлювати окремо не потрібно, так як вона входить в склад комплексу XAMPP.

До апаратного забезпечення безпосередньо відноситься процесор, який повинен якомога швидше працювати з програмним продуктом, тому він для нормального функціонування повинен по потужності буди на слабшим за процесор Intel I3.

На менш потужних процесорах програма буде працювати, але буде відчуватися значний дискомфорт у її роботі, що буде пов'язано із великими затратами часу. Також для повноцінної роботи програми необхідна наявність достатньої кількості оперативної пам'яті.

Мінімальна кількість необхідної оперативної пам'яті становить 1GB. Окрім оперативної пам'яті, потрібна відео карта з об'ємом пам'яті 64 MB, а також необхідно вільне місце на жорсткому диску, мінімальним об'ємом в 2 GB.

Для розгортання програмного забезпечення на локальному комп'ютері необхідно встановити веб-сервер.

Далі у веб-сервері відкрити «phpMyAdmin» - додаток для адміністрування баз даних MySQL. Авторизуватись параметрами по замовчуванню (логін –

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

«root», пароль - відсутній). Далі потрібно створити нову базу даних із назвою «pharmacy_db».

Для того, щоб сайт був із введеними даними, до вихідних кодів прикріплено файл дампу бази даних («pharmacy_db.sql»), який потрібно імпортувати до створеної БД.

Папку сайту потрібно скопіювати, у каталог вебсервера “domains/”.

Взаємодію користувача з ПК та його роботу з веб-сайтом, а також апаратно-програмні вимоги розглянуті у діаграмі розгортання додатку, що приведена на рисунку 3.15.

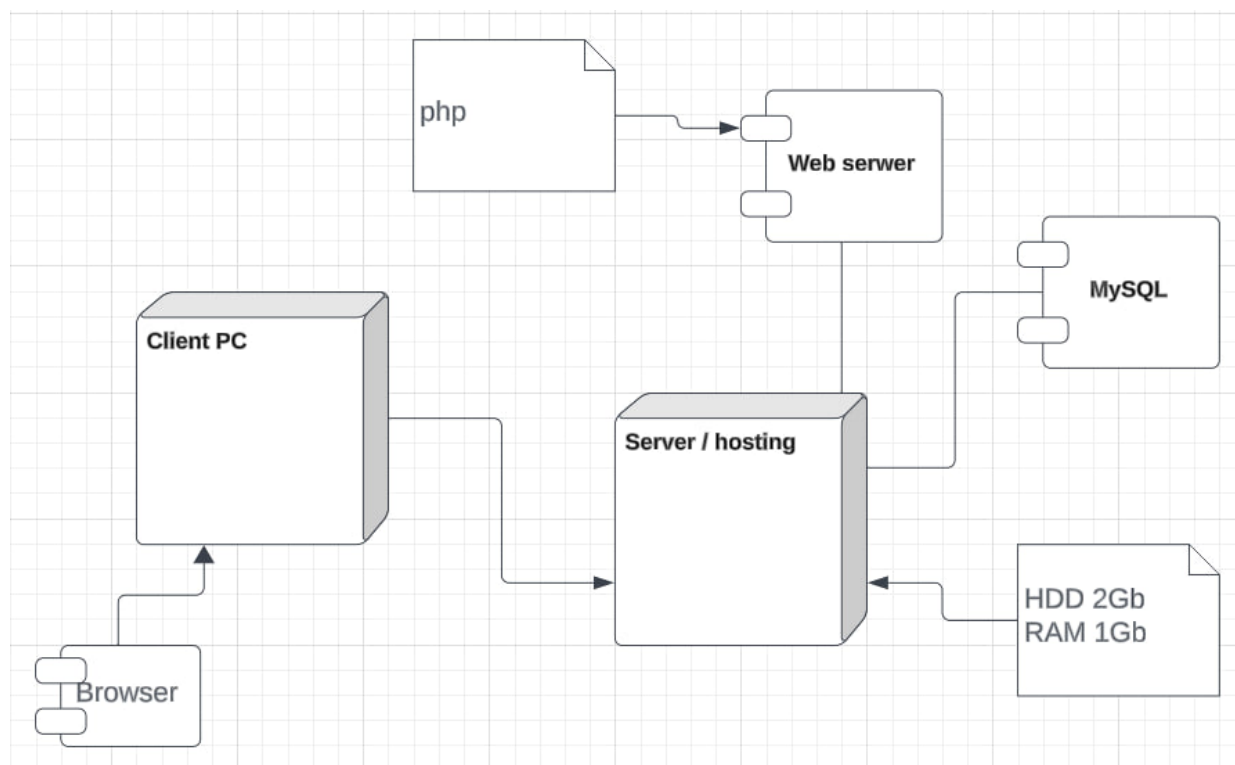


Рисунок 3.15 – Діаграма розгортання додатку

Для користування додатком спершу потрібно запустити локальний веб-сервер та відкрити розроблений сайт у встановленому на комп'ютері додатку веб-переглядачі. Для дано проєкту було вибрано веб-сервер ХАМРР.

ХАМРР - це безкоштовний, відкритий для використання і легкий у встановленні пакет програмного забезпечення, який включає в себе необхідне програмне забезпечення для створення та роботи з веб-сайтами на вашому локальному комп'ютері.

Спершу потрібно наповнити сайт вмістом, тобто додати необхідну інформацію. Для цього слід використати адміністративну частину сайту, перейшовши за посиланням <http://farmacy.lrvl/admin>. Для авторизації слід вказати логін – admin та пароль – admin (рисунок 3.16).

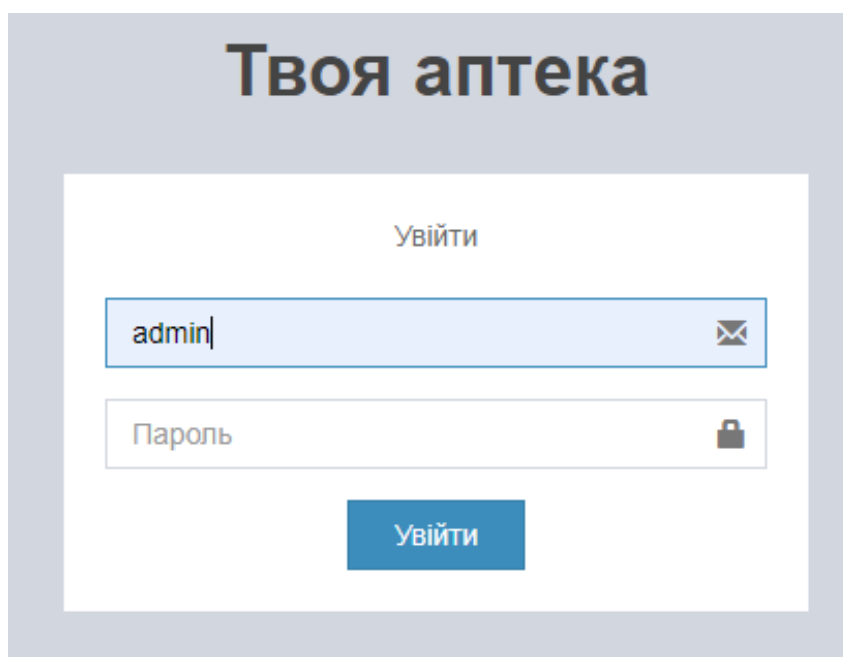


Рисунок 3.16 – Авторизація в адміністративній частині сайту

Далі можна перевірити та керувати зареєстрованими користувачами в системі (рисунок 3.17).

ID	ІМ'Я	Email	Телефон	Дата реєстрації	Опції
47	Коваленко Артем	test5@gmail.com	+380(99)090-9090	2024-03-07 20:44:14	[іконки]
44	Петренко Іван	test1@gmail.com	+380(99)809-8989	2024-03-07 20:40:10	[іконки]
45	Петренко Анатолій	test2@gmail.com	+380(99)809-8989	2024-03-07 20:42:17	[іконки]
46	Іванчук Анна	test3@gmail.com	+380(99)909-0998	2024-03-07 20:43:07	[іконки]

Рисунок 3.17 – Зареєстровані користувачі додатку

У Laravel робота з сутностями зазвичай відбувається через використання моделей Eloquent. Модель Eloquent - це PHP-клас, який представляє таблицю в базі даних. Laravel забезпечує простий та зручний спосіб роботи з базою даних через ці моделі.

Визначення таблиці та масиву fillable: В моделі ви вказуєте назву таблиці, з якою вона пов'язана, а також поля, які можна масово присвоювати (fillable). Також можна вказати інші властивості, такі як зв'язки з іншими моделями, правила валідації тощо.

Загалом, для роботи з окремими сутностями додатку в адміністративній панелі передбачене меню, яке вже є автоматично вбудоване у даний фреймворк, що дає чудову нагоду проводити додавання, редагування, оновлення та видалення даних із нашої зашальної бази даних. (рисунок 3.18).

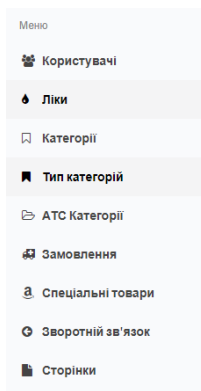


Рисунок 3.18 – Меню адміністративної панелі

У Laravel адміністративна панель зазвичай реалізується за допомогою сторонніх пакетів або власноручно шляхом розробки спеціального функціоналу. Вибір залежить від вашого завдання, потреб вашого проекту і рівня складності, який ви хочете досягти.

Замість використання сторонніх пакетів, ви можете інтегрувати готовий адміністративний шаблон, такий як AdminLTE, з Laravel. Це може бути корисно, якщо вам потрібен додатковий контроль над дизайном та функціоналом.

Якщо жоден із наявних пакетів не задовольняє ваші потреби, ви завжди можете створити власну адміністративну панель у Laravel. Для цього вам потрібно буде створити маршрути, контролери та шаблони для керування вашими даними та функціоналом.

Як приклад перейдемо до пункту меню ліки та додамо нові ліки (рисунок 3.19).

Рисунок 3.19 – Додавання нових ліків

Далі вкажемо їх форму випуску та іншу інформацію (рисунок 3.20).

Рисунок 3.20 – Додавання інформації про форму випуску нових ліків

Після додання інформації та натискання кнопки «Зберегти» бачимо, що внесені дані успішно збережені (рисунок 3.21).

ID	Код товару	Назва	Виробник	Категорія	Опції
53	3573E	Медіпор+	ЗМ	Переє'язочні засоби	

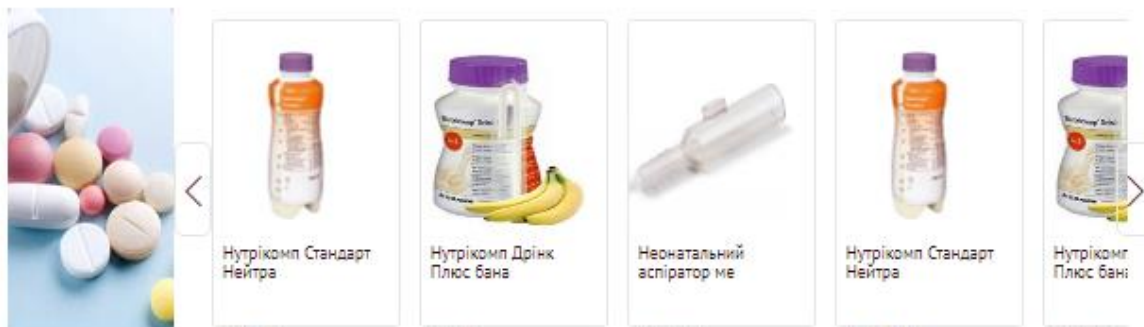
Рисунок 3.21 – Успішне додання нових ліків






Далі потрібно заповнити усі інші пусті таблиці, для того, щоб можна було вже зручно і повноцінно почати користуватись нашим веб-додатком, адже далі ми безпосередньо будемо працювати з нашим додатком.

Після наповнення вмістом можна перейти до безпосереднього використання додатку перейшовши за посиланням <http://farmacy.lrv1>, після чого з'явиться головна сторінка додатку (рисунок 3.22).

Актуальний товар

Лікарські засоби і вироби медичного призначення. Лише сертифікована продукція в Хмельницькому і по всій Україні.




	 Нутрікомп Стандарт Нейтра 121.74грн	 Нутрікомп Дрінк Плюс бана 69.26грн	 Неонатальний аспіратор ме 154.32грн	 Нутрікомп Стандарт Нейтра 121.74грн	 Нутрікомп Плюс бана 69.26грн
--	---	--	--	---	--

[Показати більше](#) ➔

Акції

Лікарські засоби і вироби медичного призначення. Лише сертифікована продукція в Хмельницькому і по всій Україні.



	 Захисний крем Молікар Скі 190грн	 Пед пов'язка адгезив. для 28.99грн	 Захисний крем Молікар Скі 190грн	 Пед пов'язка адгезив. для 28.99грн
--	--	--	--	--

Рисунок 3.22 – Головна сторінка додатку

Подальше користування додатком інтуїтивно зрозуміло. Для цього слід користуватись головним меню сайту, панеллю керування, іншими візуальними елементами інтерфейсу та підказками.

Наприклад, для пошуку та підбору потрібних лікарських засобів слід перейти за посиланням головного меню «Лікарські засоби» (рисунок 3.23).

Аналогічно можна користуватись й іншими вкладками, посиланнями додатку.

ЛІКАРСЬКІ ЗАСОБИ
КОСМЕТИКА
ДИТЯЧІ ТОВАРИ
МЕДТЕХНІКА
ВІТАМІНИ І БАДИ

Головна > Всі товари

Всі товари

Лікарські засоби ▾

Косметика ▾

Дитячі товари ▾

Медтехніка ▾

Вітаміни і БАДи ▾

Сортувати за: Від дешевих до дорогих ▾

Ціна ▲

0 - 137

▬

▬

Форма випуску ▲

Упаковка


Пластина

Капсули

Коробка


Країна виробник ▾

Умови відпуску ▾




Спрей охолоджувальний Nexcare Cold Spray...
ЗМ

91.72 грн
В наявності



Нутрікомп Стандарт Нейтральний у пластик...
BBraun


121.74 грн
В наявності



Нутрікомп Дрінк Плюс банан
BBraun


69.26 грн
В наявності

♡
ДО КОШИКА



Захисний крем з оксидом цинку Молкар Ск...
Hartmann

136.04 грн
В наявності



Медпор -
ЗМ

28.99 грн 32.9 грн
В наявності

-12%

Рисунок 3.23 – Сторінка для пошуку та підбору лікарських засобів

3.4 Висновки

У цьому розділі описаний хід програмної реалізації програмного засобу починаючи від встановлення платформи, розгортання міграцій, створення моделей, контролерів, представлень, та завершуючи проведенням тестування для визначення ступеня готовності програмного забезпечення для подальшої експлуатації.

Також була складена інструкція з інсталяції та експлуатації програмного продукту, що має полегшити відповідні дії користувачів.

ВИСНОВКИ

Під час виконання дипломного проєкту була розроблена інформаційна система для підбору товарів медичного призначення.

Дослідження предметної області дозволило визначитись із вхідними та результуючими сутностями, атрибутами проєкту, провести огляд та аналіз існуючих рішень, методів та засобів вирішення задачі, на основі чого здійснено постановку задачі дипломного проєкту.

Далі, у наступному розділі відбулось комплексне проєктування архітектури, структур даних, зокрема бази даних, інтерфейсу інформаційної системи.

Наступним кроком стала програмна реалізація та тестування проєктних рішень.

Для цього були використані наступні web-технології: за основу взято платформу Laravel; програмна реалізація здійснювалась засобами мови програмування PHP; СУБД MySQL.

Виконано успішне комплексне тестування програмного забезпечення дозволило забезпечити потрібний рівень якості додатку, безпомилкове його функціонування.

Документування інструкції з інсталяції та експлуатації робить подальше впровадження та використання додатку більш зручним та зрозумілим.

В цілому, реалізований веб додаток, інформаційна система для підбору товарів медичного призначення може практично використовуватись аптечними мережами.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		70

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Онлайн ресурс, довідник із АТС-класифікацією лікарських препаратів. URL: <https://compendium.com.ua/uk/klasyfikatsiyi/> (дата звернення: 12.02.2024)
2. Довідник з HTML та CSS. URL: <https://css.in.ua/> (дата звернення: 24.04.2024)
3. Bootstrap. URL: <https://getbootstrap.com/> (дата звернення: 24.04.2024)
4. Міжплатформовий PHP-фреймворк для створення веб-додатків, документація Laravel. URL: <https://laravel.com/> (дата звернення: 24.04.2024)
5. Інтелектуальний редактор для PHP, HTML і JavaScript PhpStorm. URL: <https://www.jetbrains.com/> (дата звернення: 24.04.2024)
6. Локальний веб сервер XAMPP. URL: <https://www.apachefriends.org> (дата звернення: 24.04.2024)
7. PHP Documentation. URL: <https://www.php.net> (дата звернення: 24.04.2024)
8. Олександр Марголін. UML для бізнес-моделювання: для чого потрібні діаграми процесів. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html> (дата звернення: 4.03.2024)
9. Основи UML. URL: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-basics.html> (дата звернення: 4.03.2024)
10. Довідкова інформація по UML (Unified Modelling Language). URL: https://uk.wikipedia.org/wiki/Unified_Modeling_Language (дата звернення: 4.03.2024)
11. Інтернет аптека 911. URL: <https://apteka911.ua> (дата звернення: 12.02.2024)
12. Інтернет аптека apteka24.ua. URL: <https://www.apteka24.ua/> (дата звернення: 12.02.2024)
13. Інтернет аптека tabletki.ua. URL: <https://tabletki.ua/> (дата звернення: 12.02.2024)

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

14. Володимир Шайтан «Побудова динамічних вебзастосунків за допомогою MVC». URL: <https://blog.ithillel.ua/articles/building-dynamic-web-applications-using-mvc> (дата звернення: 24.04.2024)

15. PHP підручник. URL: <https://w3schoolsua.github.io/php/index.html>

16. Документація та необхідні ресурси фреймворку laravel. URL: <https://laravel.com/> (дата звернення: 24.04.2024)

17. Ковальчук О.М. Інформаційні системи управління товарними запасами підприємств. Київ: Видавництво НУХТ, 2019. 298 с.

18. Петренко В.П. Технології реалізації електронних магазинів з інтегрованими системами пошуку товарів. Львів: Видавництво ЛНУ, 2020. 340 с.

19. Іванова Т.С. Аналіз та оптимізація інформаційних систем пошуку товарів за допомогою машинного навчання. Одеса: Видавництво ОНУ, 2017. 430 с.

20. Семененко В.М. Розробка та впровадження інформаційно-аналітичної системи пошуку товарів для онлайн-магазинів. Київ: Видавництво КНТЕУ, 2018. 270 с.

21. Гаврилюк І.В. Методи та алгоритми пошуку товарів у великих онлайн-маркетплейсах. Львів: Видавництво ЛНУ, 2016. 220 с.

22. Lee H., Cho J., Park S. A personalized recommendation system based on user purchase history for mobile commerce. *Expert Systems with Applications*. 2019. Vol. 132. P. 202-212.

23. Wang C., Zhang H., Zhang Z. A smart product search engine based on natural language processing. *Multimedia Tools and Applications*. 2018. Vol. 77, No 21. P. 28417-28434.

24. Chen C., Wang W., Lin C. A personalized product recommendation system based on hybrid filtering techniques. *Future Generation Computer Systems*. 2018. Vol. 78. P. 133-142.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

25. Kim H., Hwang H., Yoon Y. A personalized product recommendation system based on user behavior analysis in online shopping mall. *Computers in Industry*. 2018. Vol. 97. P. 116-126

26. Павленко Д.В. Інтеграція систем пошуку товарів у розподільчі мережі з використанням IoT-технологій. Харків: Видавництво ХНУРЕ, 2019. С. 318.

27. Гончаренко В.П. Використання інформаційних технологій у системах пошуку товарів для підвищення конкурентоспроможності підприємств. Одеса: Видавництво ОНУ, 2018. 448 с.

28. Білоус А.І. Аналіз та вдосконалення методів пошуку товарів у системах електронної комерції. Київ: Видавництво КПІ, 2017. 244 с.

29. Козлов І.О. Розробка та впровадження інформаційно-аналітичних систем пошуку товарів для онлайн-магазинів з використанням Big Data технологій. Львів: Видавництво ЛНУ, 2019. 360 с.

30. Oliver Gray., Daniel Jackson., *International Journal of Electronic Commerce. Computer technologies* 2018. Vol. 22, No. 4. P. 152.

31. Wilson G., Martinez H., Anderson J. Improving Product Search Relevance in Online Retail Using Natural Language Processing. *Journal of Retailing*. 2020. Vol. 36, No. 2. P. 112-125.

32. Thompson K., Garcia M., Lee R. Deep Learning Techniques for Enhancing Product Recommendation Systems. *IEEE Transactions on Big Data*. 2019. Vol. 5, No. 1. P. 203-215.

33. Evans S., Collins P., Murphy L. A Framework for Evaluating Product Search Algorithms in E-commerce. *Journal of Computer Science and Technology*. 2018. Vol. 11, No. 2. P. 78-91.

34. Carter R., Parker K., Hughes D. Analyzing User Behavior Patterns to Improve Search Performance in Online Marketplaces. *International Journal of Human-Computer Interaction*. 2019. Vol. 27, No. 3. P. 321-335.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

35. Adams J., Lopez M., Smith R. Enhancing Product Search Efficiency Using Neural Network Models. *Journal of Information Science*. 2018. Vol. 15, No. 4. P. 189-202.

36. White L., Taylor N., Hill S. An Empirical Study of User Satisfaction with E-commerce Search Interfaces. *ACM Transactions on Internet Technology*. 2020. Vol. 8, No. 2. P. 154-167.

37. King J., Scott L., Garcia P. Using Collaborative Filtering to Improve Product Recommendations in E-commerce. *Journal of Interactive Marketing*. 2019. Vol. 25, No. 1. P. 56-69.

38. Коваленко О.В. Розробка та впровадження інтелектуальної системи пошуку товарів на основі аналізу відгуків користувачів. *Комп'ютерні системи та мережі*. Харків, 2020. С 45-52.

39. Петренко В.М. Ефективність застосування алгоритмів машинного навчання у системах пошуку товарів. *Інформаційні технології та програмування*. Львів, 2018. С. 76-84.

40. Гриценко О.С. Вплив персоналізації на ефективність пошукових систем в електронній комерції. *Комп'ютерні науки та інформаційні технології*. Київ, 2019. С. 98-105.

41. Шевчук Н.П. Методи та алгоритми підвищення релевантності пошукових систем у веб-магазинах. *Інформаційні технології та комп'ютерна інженерія*. Львів, 2020. С. 115-122.

42. Кулик О.І. Використання глибокого навчання для покращення пошукових систем у сфері електронної комерції. *Комп'ютерна інженерія та інформаційні технології*. Київ, 2018. С. 132-139.

43. Мельник В.О. Розробка інтегрованої системи пошуку товарів для онлайн-магазину. *Інформаційні технології та комп'ютерна інженерія*. Харків, 2019. С. 87-94.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

44. Павленко І.М. Вплив використання рекомендаційних систем на покращення пошукових функцій в електронній комерції. *Інформаційні технології та програмування*. Львів, 2018. С. 105-112.

45. Ковальчук М.В. Оптимізація алгоритмів ранжування результатів пошуку в електронних магазинах. *Комп'ютерні науки та інформаційні технології*. Київ, 2020. С. 76-83.

46. Іванова Н.І. Використання технологій обробки природної мови для покращення результатів пошукових систем у веб-магазинах. *Інформаційні технології та комп'ютерна інженерія*. Львів, 2019. С. 98-105.

47. Кравченко В.М. Інформаційні системи підтримки прийняття рішень у сфері управління товарними запасами. Одеса: Видавництво ОНПУ, 2017. 192 с.

48. Шевченко Л.С. Автоматизовані системи пошуку товарів для онлайн-ритейлу. Київ: Видавництво КНТЕУ, 2019. 188 с.

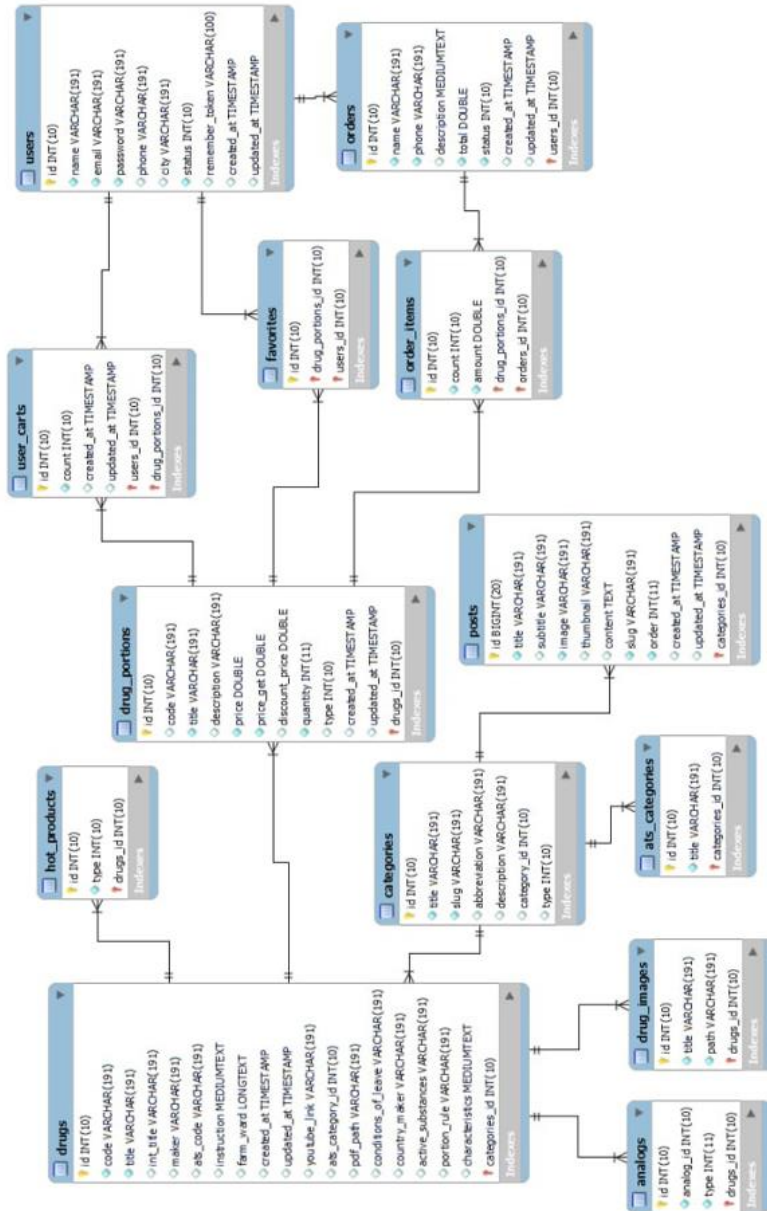
49. Ю.О. Інтеграція систем пошуку товарів у веб-сайти малого бізнесу. Львів: Видавництво ЛНУ, 2018. 214 с.

					КВРІСТ.101041. 21.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		75

ДОДАТОК А (обов'язковий) Структура бази даних

КвРІСТ.101041.21.01.03 ІТЗ

Схема бази даних

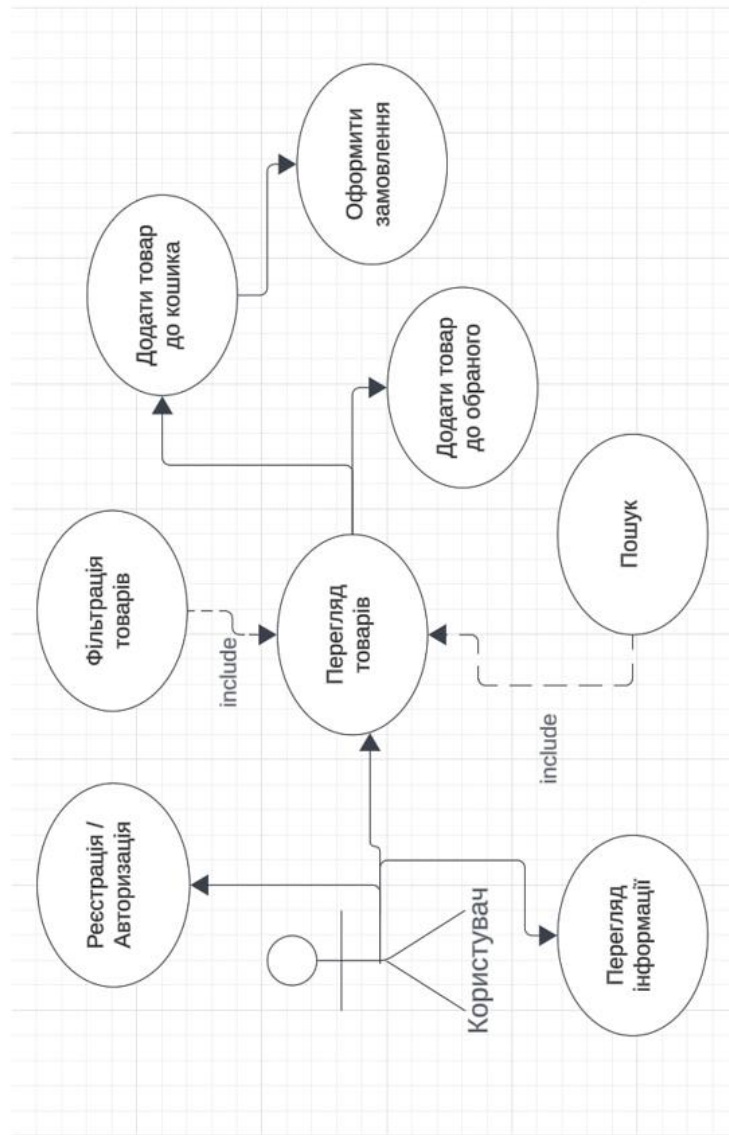


КвРІСТ.101041.21.01.03 ІТЗ			
Інформаційна система підбору товарів			
Зам. Аск.	№ докум.	Підпис	Дата
Розроб.	Виконав.		
Н. Коштр.	І. Коштр.		
Т. Коштр.	І. Коштр.		
Зам.	Розробник		
		Листопад	Месець
		Август 1	Август 3
		ХНУ, ГР.ІСТ-21-1	

ДОДАТОК Б (обов'язковий)

UML діаграма варіантів дій інформаційної системи підбору товару

UML діаграма варіантів дій

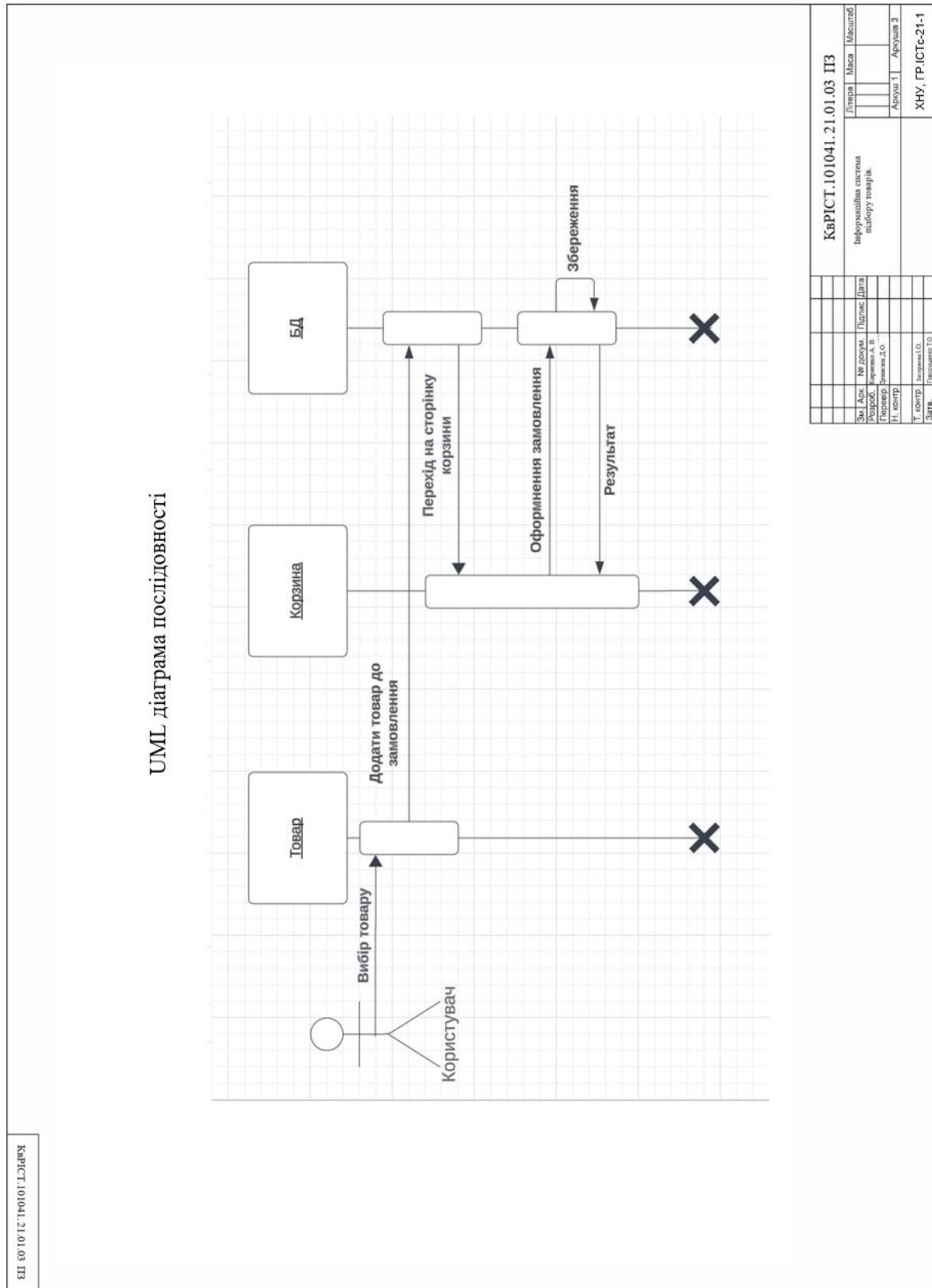


КАРІСТ.101041.21.01.03 ПЗ

КАРІСТ.101041.21.01.03 ПЗ			
Зм. Акц.	№ докум.	Підпис	Дата
Розроб.	Журило А. В.		
Перевір.	Данюк Д.О.		
Н. контр.			
Т. контр.	Засурман І.О.		
Зам.б.	Горюханов Г.О.		
		Літера	Місяц
		Інформаційна система підбору товарів	
		Аркуш 1	Аркуш 3
		ХНУ, ГР.ІСТс-21-1	

ДОДАТОК В (обов'язковий)

UML діаграма послідовності дій інформаційної системи підбору товару



ДОДАТОК Г

«Лістинг інформаційної системи підбору товару»

```
<?php
```

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateDrugsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('drugs', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('code')->unique();
            $table->string('title');
            $table->string('int_title')->nullable();
            $table->string('maker')->nullable();
            $table->string('ats_code')->nullable();
            $table->mediumText('instruction')->nullable();
            $table->string('farm_ward')->nullable();
            $table->json('analogs')->nullable();
            $table->unsignedInteger('category_id')->nullable();
            $table->timestamps();
        });

        Schema::create('drug_portions', function (Blueprint $table) {
            $table->increments('id');
            $table->unsignedInteger('drug_id');
            $table->string('code')->unique();
            $table->string('title');
            $table->string('description')->nullable();
            $table->double('price');
            $table->double('price_get');
            $table->integer('quantity');
            $table->unsignedInteger('type')->nullable();
        });
    }
}
```

```

        $table->timestamps();

        $table->foreign('drug_id')->references('id')->on('drugs')->onDelete('cascade');
    });

    Schema::create('drug_images', function (Blueprint $table) {
        $table->increments('id');
        $table->unsignedInteger('drug_id');
        $table->string('title');
        $table->string('path');

        $table->foreign('drug_id')->references('id')->on('drugs')->onDelete('cascade');
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('drugs');
    Schema::dropIfExists('drug_portions');
    Schema::dropIfExists('drug_images');
}
}
}

<?php

namespace App\Http\Controllers;

use App\Models\Analog;
use App\Unit;
use App\Models\Category;
use App\Models\Drug;
use App\Models\DrugPortion;
use App\Models\HotProducts;
use App\Repositories\DrugRepository;
use App\Repositories\UserRepository;
use Illuminate\Http\Request;

```

```

use Illuminate\Support\Facades\Input;
use Illuminate\Pagination\Paginator;

class DrugController extends Controller
{
    /**
     * @var DrugRepository
     */
    protected $drug;

    /**
     * @var UserRepository
     */
    protected $user;

    /**
     * DrugController constructor.
     *
     * @param DrugRepository $drug
     * @param UserRepository $user
     */
    public function __construct(DrugRepository $drug, UserRepository $user)
    {
        $this->drug = $drug;
        $this->user = $user;
    }

    /**
     * @param $code
     *
     * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
     */
    public function show($code)
    {
        $drug = $this->drug->find($code);
        $drug = $drug->drug();
        $portions = $this->drug->getAllPortions($drug);

        /**
         * Paginate portions for 1-2 and other
         */
    }
}

```

```

    $firstPortions = null;
    $lastPortions = null;

    if (isset($portions[0])) {
        $firstPortions[] = $portions[0];
    }

    if (isset($portions[1])) {
        $firstPortions[] = $portions[1];
    }

    if (isset($portions[2])) {
        unset($portions[0], $portions[1]);

        $lastPortions = $portions;
    }
    $actualProducts = $this->drug->getActualProductsQuery()->take(8)->get();

    return view('pages.product_card', compact('drug', 'firstPortions', 'lastPortions',
'actualProducts'));
}

/**
 * @param $code
 *
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function showCategory($code, Request $request)
{
    $formType = null;
    $makerType = null;
    $rating = null;

    $category = Category::where('id', $code)->first();

    $categories = Category::getAll($category);
    array_push($categories, $category->id);

    $products = DrugPortion::whereHas('drug', function ($query) use ($categories) {
        $query->whereIn('category_id', $categories);
    });
}

```

```

$category = $category->title;

if (isset($products)) {
if (isset($products) && request()->get('formType') && request()->get('makerType')
&& request()->get('rating'))
{
    $formType = request()->get('form');
    if($formType)
        $formType = explode(',', $formType);

    $makerType = request()->get('makerType');
    $rating = request()->get('rating');
    $page = request()->get('page');
    $price1 = request()->get('price-from');
    $price2 = request()->get('price-to');

    $country = request()->get('country');
    if($country)
        $country = explode(',', $country);

    $limitations = request()->get('limitations');
    if($limitations)
        $limitations = explode(',', $limitations);

    $max_price = $products->max('price');

    /**
     * FormTypes filter
     */
    if ($formType !== 'default' && $formType !== null) {
        $products = $products->whereIn('type',$formType);
        $products = $products->whereType($formType);
    }

    /**
     * Maker filter
     */
    if ($makerType !== 'default' && $makerType !== null) {
        $products = $products->whereHas('drug', function ($query) use ($makerType)
{

```

```

        $query->whereMaker($makerType);
    });
}

/**
 * Sort filter
 */
if ($rating !== 'default' && $rating !== null) {
    if ($rating === 'price-cheap') {
        $products = $products->orderBy('price', 'asc');
    } elseif ($rating === 'price-costly') {
        $products = $products->orderBy('price', 'desc');
    } elseif ($rating === 'availability') {
        $products = $products->orderBy('quantity', 'desc');
    }
}

/**
 * Price filter
 */
if ($price1 !== null) {
    $products = $products->where('price','>=',$price1);
    $products = $products->where('price','<=',$price2);
}

/**
 * Country filter
 */
if ($country !== 'default' && $country !== null) {
    $products->whereHas('drug', function ($query) use ($country) {
        $query->whereIn('country_maker',$country);
    });
}

/**
 * Limitations filter
 */
if ($limitations !== 'default' && $limitations !== null) {
    $products->whereHas('drug', function ($query) use ($limitations) {
        $query->whereIn('conditions_of_leave',$limitations);
    });
}

```

```

    }
}

if (isset($products)) {
    /**
     * Get Paginated products
     */

    Paginator::currentPageResolver(function () use ($page) {
        return $page;
    });

    $products_count = $products->count();
    $products = $products->paginate(9);

    if(isset($request->load)){
        $html = view('layouts.list_product_elements')->with(compact('products'))->render();
        return response()->json(['success' => true, 'html' => $html, 'products_count' => $products_count, 'page' => $rating]);
    }

    return view('pages.catalog', [
        'products' => $products->appends(Input::except('page')),
        'category' => $category,
        'formType' => $formType,
        'makerType' => $makerType,
        'rating' => $rating,
        'products_count' => $products_count,
        'max_price' => $max_price,
        'formt' => $formType,
        'limitations' => $limitations
    ]);
}

return view('exceptions.not-found');

}
}
<?php

```

```

namespace App\Http\Controllers;

use App\Models\Category;
use App\Models\CategoryType;
use App\Models Drug;
use App\Models DrugPortion;
use App\Models Post;
use App\Repositories DrugRepository;
use Auth;
use Illuminate\Http Request;
use Illuminate\Pagination\Paginator;
use Illuminate\Support\Facades Input;

class PageController extends Controller
{
    /**
     * @var DrugRepository $drugs
     */
    protected $drugs;

    /**
     * Word for search
     *
     * @var $findWord
     */
    protected $findWord = null;

    /**
     * PageController constructor.
     * @param DrugRepository $drugs
     */
    public function __construct(DrugRepository $drugs)
    {
        $this->drugs = $drugs;
    }

    public function index()
    {
        $actualProducts = $this->drugs->getActualProductsQuery()->take(8)->get();
        $discountProducts = $this->drugs->getDiscountProductsQuery()->take(8)->get();
    }
}

```

```

    return view('pages.index', compact('actualProducts', 'discountProducts'));
}

public function search($word, Request $request)
{
    $this->findWord = $word;

    return $this->show('search', $request);
}

public function searchbar(Request $request){
    $this->findWord = $request->all()['text'];
    $products = $this->drugs->getSearchQuery($this->findWord)->limit(4)->get();

    $html = view('layouts.search_list_product_elements')->with(compact('products'))-
    >render();

    return response()->json(['success' => true, 'html' => $html]);
//    return compact('products');
}

/**
 * @param $page
 * @param Request $request
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function show($page, Request $request)
{
    $category = null;
    $formType = null;
    $makerType = null;
    $rating = null;

    switch ($page) {
        /**
         * Categories
         */
        case 'likarski_zasoby':
        case 'kosmetyka':
        case 'dityachi_tovary':
        case 'medtekhnika':

```

```

case 'vitaminy_i_bady':
    $type = CategoryType::where('url',$page)->firstOrFail();
    $category = $type->name;
    $products = $this->drugs->getByCategoryTypeQuery($type->id);
    break;
case 'actual':
    $products = $this->drugs->getActualProductsQuery();
    $category = 'Актуальні товари';

    break;
case 'popular':
    $products = $this->drugs->getPopularProductsQuery();
    $category = 'Популярні товари';

    break;
case 'discount':
    $products = $this->drugs->getDiscountProductsQuery();
    $category = 'Акційні товари';

    break;
case 'catalog':
    $products = $this->drugs->getQuery();
    $categories = Category::get()->pluck('id')->toArray();

    $products->whereHas('drug', function ($query) use ($categories) {
        $query->whereIn('category_id', $categories);
    });

    $category = 'Всі товари';

    break;
/**
 * Search
 */
case 'search':
    if ($this->findWord) {
        $category = 'Пошук по - ' . $this->findWord;

        $products = $this->drugs->getSearchQuery($this->findWord);
    } else {
        $products = $this->drugs->getQuery();
    }

```

```

    }

    break;

/**
 * Other pages
 */
case 'certificates':
    return view('pages.certificates');
default:
    $post = Post::where('slug',$page)->first();
    if($post)
        return view('pages.page_template', compact('post'));
    else
        return view('exceptions.not-found');
}

if (isset($products)) {

    $formType = request()->get('form');
    if($formType)
        $formType = explode(',', $formType);

    $makerType = request()->get('makerType');
    $rating = request()->get('rating');
    $page = request()->get('page');
    $price1 = request()->get('price-from');
    $price2 = request()->get('price-to');

    $country = request()->get('country');
    if($country)
        $country = explode(',', $country);

    $limitations = request()->get('limitations');
    if($limitations)
        $limitations = explode(',', $limitations);

    $max_price = $products->max('price');

/**
 * FormTypes filter

```

```

*/
if ($formType !== 'default' && $formType !== null) {
    $products = $products->whereIn('type',$formType);
}

/**
 * Maker filter
 */
if ($makerType !== 'default' && $makerType !== null) {
    $products = $products->whereHas('drug', function ($query) use ($makerType)
{
        $query->whereMaker($makerType);
    });
}

/**
 * Sort filter
 */
if ($rating !== 'default' && $rating !== null) {
    if ($rating === 'price-cheap') {
        $products = $products->orderBy('price', 'asc');
    } elseif ($rating === 'price-costly') {
        $products = $products->orderBy('price', 'desc');
    } elseif ($rating === 'availability') {
        $products = $products->orderBy('quantity', 'desc');
    }
}

/**
 * Price filter
 */
if ($price1 !== null) {
    $products = $products->where('price','>=',$price1);
    $products = $products->where('price','<=',$price2);
}

/**
 * Country filter
 */
if ($country !== 'default' && $country !== null) {
    $products->whereHas('drug', function ($query) use ($country) {
        $query->whereIn('country_maker',$country);
    });
}

```

```

    });
}
/**
 * Limitations filter
 */
if ($limitations !== 'default' && $limitations !== null) {
    $products->whereHas('drug', function ($query) use ($limitations) {
        $query->whereIn('conditions_of_leave',$limitations);
    });
}
/**
 * Get Paginated products
 */
Paginator::currentPageResolver(function () use ($page) {
    return $page;
});

$products_count = $products->count();
$products = $products->paginate(9);

if(isset($request->load)){

    $html = view('layouts.list_product_elements')->with(compact('products'))->render();

    return response()->json(['success' => true, 'html' => $html, 'products_count' => $products_count, 'page' => $rating]);

}
return view('pages.catalog', [
    'products' => $products->appends(Input::except('page')),
    'category' => $category,
    'formType' => $formType,
    'makerType' => $makerType,
    'rating' => $rating,
    'products_count' => $products_count,
    'max_price' => $max_price,
    'formt' => $formType,
    'limitations' => []
]);
}

```

```

        return view('exceptions.not-found');
    }
<?php

/**
 * Main page
 */
Route::get('/', 'PageController@index')->name('pages.index');
Route::get('/mail', function () {
    return view('mail.register');
});

/**
 * Product page
 */
Route::get('product/{code}', 'DrugController@show')->name('product.show');
Route::get('category/{code}', 'DrugController@showCategory')->name('category.show');

/**
 * Product search
 */
Route::get('search/{word}', 'PageController@search')->name('product.search');
Route::post('search', 'PageController@searchbar')->name('searchbar');

/**
 * Cart routes
 */
Route::get('cart', 'CartController@show')->name('cart.show');
Route::post('cart', 'CartController@add')->name('cart.add');
Route::get('cart/clear', 'CartController@clear')->name('cart.clear');

Route::post('cart/change', 'CartController@changeQuantity')->name('cart.change');
Route::post('cart/remove', 'CartController@remove')->name('cart.remove');

/**
 * User routes
 */
Route::post('user/general', 'ProfileController@changeGeneral')->name('user.change.general');

```

```

Route::post('user/email', 'ProfileController@changeEmail')->
name('user.change.email');
Route::post('user/password', 'ProfileController@changePassword')->
name('user.change.password');

/**
 * User Cabinet page
 */
Route::get('cabinet', 'ProfileController@show')->name('cabinet.show');

/**
 * Make order
 */
Route::post('order/make', 'OrderController@make')->name('order.make');

/**
 * Classification
 */
Route::get('classification', 'CategoryController@classificationShow')->
name('classification.show');
Route::get('classification/{id}', 'CategoryController@classificationShow')->name('sub-
classification.show');

/**
 * Feedback
 */
Route::post('feedback', 'FeedbackController@create')->name('feedback.make');

/**
 * Favorites
 */
Route::post('favorite', 'FavoriteController@add')->name('favorite.add');
Route::post('favorite/remove', 'FavoriteController@remove')->
name('favorite.remove');

/**
 * Pages controller route
 */
Route::get('/{page}', 'PageController@show')->name('pages.page');

```

«ЛІСТИНГ unit-ТЕСТІВ»

```
<?php
namespace Tests\Unit;
use Tests\TestCase;
use App\Models\Drug;
use App\Models\DrugPortion;

class DrugTest extends TestCase
{
    public function testExample()
    {
        echo "Start testing ... \n";
        $this->assertTrue(true);
        echo ".simple test .... OK \n";
    }
    public function test_drug_attr()
    {
        $drug = new Drug(["code" => "test_code", "title" => "test_title"]);
        $this->assertTrue($drug->code == "test_code");
        $this->assertFalse($drug->title == "title");
        echo "drug_attr test .... OK \n";
    }
    public function test_drug_and_portion_inserting()
    {
        $drug = new Drug(["id" => 600001, "code" => "test_code",
"title" => "test_title"]);
        $portion = new DrugPortion(["drug_id" => 600001, "title" => "portion_title",
"price"=>11, "price_get"=>9, "quantity"=>100]);
        $this->assertContains($portion, $drug->portions->add($portion));
        echo "test_drug_and_portion_inserting .... OK \n";
    }
    public function test_drug_and_portion_remove()
    {
        $drug = new Drug(["id" => 700003, "code" => "test_code",
"title" => "test_title"]);
        $portion = new DrugPortion(["drug_id" => 700003, "title" => "portion_title",
"price"=>11, "price_get"=>9, "quantity"=>100]);
        $drug->portions->add($portion);
        $this->assertTrue( $drug->portions[0]->title == "portion_title" );
        unset($drug->portions);
        $this->assertFalse( isset($drug->portions[0]));
    }
}
```

```

        echo "test_drug_and_portion_remove .... OK\n";
    }
}

```

```

<?php
namespace Tests\Feature;
use Tests\TestCase;
use Illuminate\Foundation\Testing\WithFaker;
use Illuminate\Foundation\Testing\RefreshDatabase;
class pagesTest extends TestCase
{
    public function testExample()
    {
        echo "Start testing ...\n";
        $this->assertTrue(true);
    }
    public function test_main_page()
    {
        $response = $this->get('/');
        $response->assertStatus(200);
        echo "test_main_page .... OK\n";
    }
    public function test_catalog()
    {
        $response = $this->get('/catalog');
        $response->assertStatus(200);
        $response->assertSee('Всі товари');
        $response->assertSee('<span>Всі товари</span>');
        echo "test_catalog .... OK\n";
    }
    public function test_cart_add()
    {
        $response = $this->json('POST', '/cart', ['code' => 107]);
        $response->assertStatus(302);
        echo "test_cart_add .... OK\n";
    }
    public function test_serchbar()
    {
        $response = $this->json('POST', '/search', ['text' => "Спрей"]);
        $response->assertJsonStructure([
            "success",

```

```
    "html"  
  });  
  echo "test_serchbar ... OK\n";  
}  
}
```

Ім'я користувача:
Кафедра КІ

Дата перевірки:
07.06.2024 17:58:12 EEST

Дата звіту:
07.06.2024 18:06:54 EEST

ID перевірки:
1016333071

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100005591

Назва документа: Кириєнко_Інформаційна система підбору товарів

Кількість сторінок: 77 Кількість слів: 9038 Кількість символів: 71642 Розмір файлу: 4.70 MB ID файлу: 1016133141

12.6% Схожість

Найбільша схожість: 3.75% з джерелом з Бібліотеки (ID файлу: 1016117190)

9.47% Джерела з Інтернету 415

7.58% Джерела з Бібліотеки 154

0.28% Цитат

Цитати 1

Не знайдено жодних посилань

0% Вилучень

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 10

Підозріле форматування 21 сторінка

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 3.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилوک в документах: 12%**

ID: 129096 Назва: БКР Інформаційна система підбору товарів Додано в БД: 2024-06-07 Автора: А. В. Кириєнко Керівники: Д. О. Денисюк Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	60330	563	5751 (10%)	63 (11%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Кириєнко Анатолій Володимирович

Тема: Інформаційна система пошуку товарів

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 70

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є проєктування та програмна реалізація інформаційної система підбору товарів.
2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.
3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі кваліфікаційної роботи проведено дослідження предметної області, аналіз вхідних і вихідних даних (проаналізовано існуючі аналоги інформаційних систем, їх переваги та недоліки) та виконано постановку задачі дослідження. В другому розділі кваліфікаційної роботи проведено моделювання та проєктування інформаційної системи, а саме: виконано проєктування архітектури інформаційної системи, проєктування бази даних (визначення основних об'єктів предметної області та список усіх атрибутів) та інтерфейсу(розподілення послідовності відображення усіх компонентів інтерфейсу). В третьому розділі кваліфікаційної роботи виконано програмну реалізацію інформаційної системи, а саме: реалізовано функціональне тестування, GUI тестування, модульне тестування, атакож інсталяція і експлуатація інформаційної системи .
4. Позитивні сторони роботи: висока практична цінність роботи.
5. Негативні сторони роботи: недостатня увага при використанні інформаційної системи на мобільних пристроях.

6. Оцінка графічного оформлення та пояснювальної записки роботи:
Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

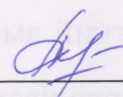
8. Інші зауваження: _____

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

доцент кафедри АКИТтаР Корецька Л.О.

“07” 06 2024 р.

 (підпис)

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система підбору товарів

Автор: Кириєнко Анатолій Володимирович

Спеціальність: 126– Інформаційні системи та технології

Освітня програма: освітньо-професійна

Науковий керівник: Денисюк Дмитро Олександрович, старший викладач

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

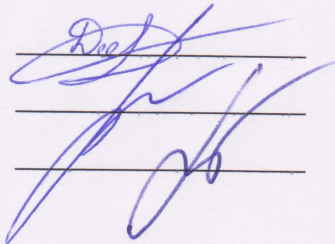
- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 12.6% і адресується до 569 першоджерела з них 9.47% джерела з інтернету та 7.58% джерела з бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



Д. О. Денисюк

Є.Г. Гнатчук

Т. О. Говорущенко