ЗМІСТ

[**ВСТУП** 2](#_Toc36429472)

[**1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ** 3](#_Toc36429473)

[**2 ПОСТАНОВКА ЗАВДАННЯ** 4](#_Toc36429474)

[**3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ** 5](#_Toc36429475)

[3.1 Виявлення основних сутностей предметної області 5](#_Toc36429476)

[3.2 Побудова схеми бази даних 8](#_Toc36429477)

[**4 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ** 10](#_Toc36429478)

[4.1 Загальний опис програмного продукту 10](#_Toc36429479)

[4.2 Опис архітектури програмної системи 13](#_Toc36429480)

[4.3 Опис концептуальної моделі бази даних 14](#_Toc36429481)

[4.4 Опис програної реалізації 20](#_Toc36429482)

[4.5 Опис задач автоматизації та інтерфейсу користувача 21](#_Toc36429483)

[**ВИСНОВКИ** 27](#_Toc36429484)

[**ПЕРЕЛІК ПОСИЛАНЬ** 28](#_Toc36429485)

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

*1*

*ККЗ. КР. 121.003. 20. ПЗ*

Розроб.

*Брусенцов Д.А.*

Керівник.

*Андрійченко Т.Р.*

Реценз.

Н. Контр.

Затверд.

*Інформаційна система*

*«Аптечний склад»*

Літ.

Акрушів

*29*

*РПЗ – 73б*

# ВСТУП

Використання баз даних є однією з характерних рис більшості сучасних інформаційних систем. По своїй суті бази даних є тим, навколо чого і будується інформаційна система будь-якого підприємства. Тому теорії створення та практиці використання баз даних приділяється достатня увага протягом періоду функціонування інформаційних систем. Досить тривалий час основним типом були реляційні бази даних, які на сьогодні вже вважаються класичними. Проте розвиток інформаційних систем поставив перед сучасними базами даних завдання, вирішення яких неможливе в межах використання тільки реляційних баз даних. Крім класичних завдань, сучасні бази даних повинні забезпечувати багатомашинну обробку та зберігання великих обсягів інформації, оперативний аналіз даних, інтеграцію із мережею Інтернет, розмежування доступу користувачів до зберігає мої інформації, захист інформації під час її передачі по мережі. Хоча на практиці і використовується чимало різноманітних баз даних, але для більшості з них існує велика кількість спільних ознак, як з погляду розробки, так і використання. Це дає можливість вивчати сучасні бази даних і відповідне прикладне та системне програмне забезпечення на прикладах, які, незважаючи на свою новизну, вже стали класичними. Як такі приклади вибрано загальні питання проектування, розробки та використання бази даних phpMyAdmin. Це пояснюється тим, що phpMyAdmin є однією із найпоширеніших і вдосконалених баз даних.

В даному курсовому проекті за предметну область взято базу даних, яка буде зберігати усю необхідну інформацію про аптечний склад. Цей додаток створений для того, щоб співробітники складу могли моніторити данні про товари, які знаходяться на складі. Система дає змогу редагувати та додавати данні про товари на складі та про клієнтів, які закуповують в них ці товари. Також база має інформацію про накладні, за допомогою яких можна отримати потрібну інформацію дуже швидко.

Команди SQL можуть виконуватись як над цілою групою таблиць, так і над одним об’єктом. Завдяки використанню SQL можливо створювати бази даних, виконувати над ними дії та робити запити до баз даних. Частіше SQL використовують для запитів. Запит - це команда, яка формується для однієї чи декількох таблиць бази даних і виводить певну інформацію з них. Запити не змінюють інформацію у таблицях, а лише показують її користувачу. Усі запити SQL формуються на базі однієї нескладної за структурою команди - SELECT. У найпростішому вигляді команда SELECT дає інструкцію базі даних для пошуку інформації у таблиці.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Головні ідеї сучасних інформаційних технологій базуються на концепції баз даних. Відповідно до цієї концепції, основою інформаційних технологій є дані, які повинні бути організовані в бази даних в цілях адекватного відображення мінливого реального світу і задоволення інформаційних потреб користувачів. Одним з найважливіших і основоположних понять в теорії баз даних є поняття інформації. Під інформацією розуміються будь-які відомості про будь-яку подію, процеси, об'єкти. У свою чергу, дані - це інформація, представлена в певному вигляді, що дозволяє автоматизувати її збір, зберігання і подальшу обробку людиною або інформаційним засобом. Для комп'ютерних технологій дані - це інформація в дискретному, фіксованому вигляді, зручна для зберігання, обробки на ЕОМ, а також для передачі по каналах зв'язку. Предметною областю даного проекту є аптечний склад. Кожну годину на склад приходить велика к-сть товарів з усіх куточків країни. Інформація про товари редагується та зберігається в базі даних. В програмі інформація може фільтруватися для комфортного перегляду та редагування. Кожен товар відноситься до певної групи товарів. За ними і можна фільтрувати препарати.

Ефективне функціонування сучасного підприємства неможливо без застосування інформаційних систем. Інформаційні системи мають ряд істотних відмінностей від стандартних прикладних програм. Залежно від предметної області інформаційні системи можуть сильно відрізнятися за своєю архітектурою і функцій. При розробці бази даних «Аптечний склад» було проведено обстеження предметної області. В організації, для якої була розроблена дана БД, вхідна інформація являє собою сукупність даних про аптеки, накладні і товари.

# 2 ПОСТАНОВКА ЗАВДАННЯ

Створити додаток для «аптечний склад». Який має наступні можливості:

* Вхід в систему відбувається зі сторони співробітників складу;
* Співробітник має змогу редагувати відомості про всі доступні таблиці в базі даних;

Додаток має бути зв’язаний з SQL сервером. Сервер має в собі містити повну інформацію в даній предметній області. Тобто потрібно створити наступні таблиці:

* goods – список товарів та їх інформація;
* goods\_group – список груп товарів;
* invoice – список накладних про перевезення товарів зі складу;
* pharmacy – список клієнтів, а саме аптек, які використовують складське приміщення;
* storage – список всіх товарів на складі, а також інформація про їхнє прибуття;

# 3 ПРОЕКТУВАННЯ БАЗИ ДАНИХ

## 3.1 Виявлення основних сутностей предметної області

Сутності:

* Накладні – invoice (Таблиця 3.3);
* Товари – goods (Таблиця 3.1);
* Клієнти(аптеки) – pharmacy (Таблиця 3.4);
* Групи товарів – goods\_group (Таблиця 3.2);
* Сховище - storage (Таблиця 3.5).

Зв’язки між сутностями:

* Товари поділяються на групи, для комфортного редагування та моніторингу;
* В накладних зображено інформацію про доставку товару.

Атрибути сутностей :

* кожна накладна має інформацію про доставку товару, товар та аптеку - клієнт;
* кожен товар має данні про групу, в якій цей товар знаходиться, назву, ціну;
* клієнти(аптеки) мають інформацію про адресу та зображення;
* кожна група товарів має назву;
* сховище має данні про товар та день, коли його завезли.

Таблиця 3.1 – Товари

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип даний | Обмеження цілісності |
| ід\_товару | Int(9) | Not Null |
| назва\_товару | Varchar(1024) | Not Null |
| ід\_групи | Int(9) | Not Null |
| ціна | Int(9) | Not Null |
| країна виробник | Varchar(1024) | Not Null |

Таблиця 3.2 – Групи товарів

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип даний | Обмеження цілісності |
| ід\_групи | Int(9) | Not Null |
| назва\_групи | Varchar(1024) | Not Null |

Таблиця 3.3 – Накладні

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип даний | Обмеження цілісності |
| ід\_накладної | Int(9) | Not Null |
| ід\_товару | Varchar(9) | Not Null |
| ід\_аптеки | Varchar(9) | Not Null |
| дата\_покупки | Date(1024) | Not Null |

Таблиця 3.4 – Клієнти(аптеки)

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип даний | Обмеження цілісності |
| ід\_аптеки | Int(9) | Not Null |
| адреса\_аптеки | Varchar(1024) | Not Null |
| фото\_аптеки | Varchar(1024) | Not Null |

Таблиця 3.5 – Сховище

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип даний | Обмеження цілісності |
| ід\_товару | Int(9) | Not Null |
| дата\_доставки | Date(1024) | Not Null |

Нормалізація відношень – покроковий процес розділення (декомпозиції) початкових відношень БД на простіші. Кроки цього процесу переводять схему відношення БД в послідовні нормальні форми. Кожна наступна форма володіє кращими властивостями ніж попередня. Кожній нормальній формі відповідає певний набір обмежень. При переведенні структури відношення у форми вищого порядку досягають видалення з таблиць надмірної описової інформації. Процес нормалізації заснований на понятті функціональної залежності атрибутів.

Перша нормальна форма. Відношення відповідає 1NF(рисунок 3.1), тоді, коли на перетині кожного стовпця і кожного рядка знаходяться тільки елементарні (неподільні) значення атрибутів і не містяться групи, що повторюються.

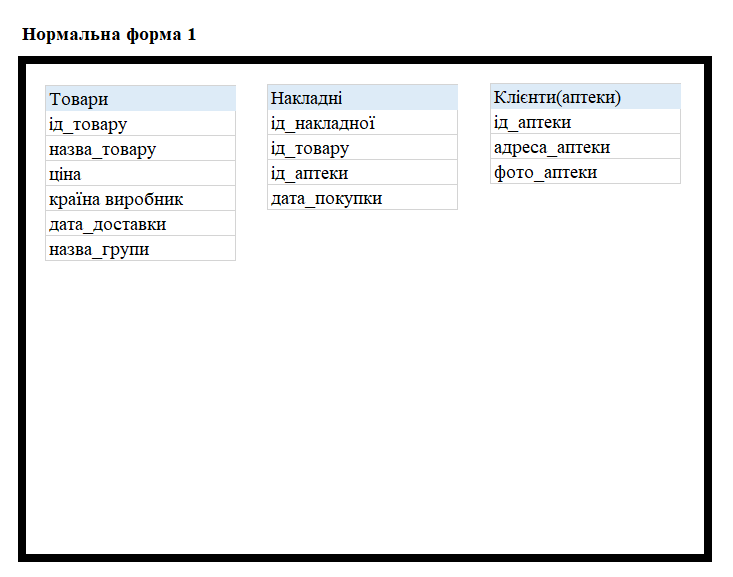


Рисунок 3.1 – Перша нормальна форма

Друга нормальна форма. Відношення знаходиться в 2NF, якщо виконуються обмеження 1NF і кожен описовий атрибут функціонально повно залежить від первинного ключа (у тому числі і складеного).

Третя нормальна форма. Відношення знаходиться у ЗNF(рисунок 3.2), якщо виконуються обмеження 2NF і всі описові атрибути відношення взаємно незалежні і повністю залежать від первинного ключа, тобто кожний описовий атрибут не транзитивно залежить від ключа.



Рисунок 3.2 – Третя нормальна форма

## 3.2 Побудова схеми бази даних

На рисунку 3.3 зображено ER-діаграму для бази даних «Аптечний склад». Модель «сутність-зв'язок» (ER-модель) — модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. ER-модель — це мета-модель даних, тобто засіб опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення, що його реалізує, є модель «сутність-зв'язок».

Важливим є той факт, що з моделі «сутність-зв'язок» можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна, об'єктна), тому вона є найзагальнішою.



Рисунок 3.3 – ER-діаграма бази даних «Аптечний склад»

# 4 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 4.1 Загальний опис програмного продукту

Даний продукт створений в середовищі Visual Studio, при створенні бази даних було використано локальний сервер OpenServer, і додаток для редагування баз даних phpMyAdmin.

Для використання продукту необхідно мати наступні характеристики апаратного забезпечення:

* Процесор - шестиядерний Intel Core i5-8400
* Обсяг оперативної пам'яті - 4 ГБ
* Відеокарта - Intel UHD Graphics 630
* Обсяг вільного місця на HDD/SSD - 36 МБ

Для написання курсової роботи нам потрібно мати сервер з повним доступом. Для наших потреб, нам буде достатньо локального сервера OpenServer.

Open Server Panel - це портативна серверна платформа і програмне середовище, створена спеціально для веб-розробників з урахуванням їх рекомендацій і побажань.

Програмний комплекс має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості з адміністрування та налаштування компонентів. Платформа широко використовується з метою розробки, налагодження і тестування веб-проектів, а так само для надання веб-сервісів в локальних мережах.

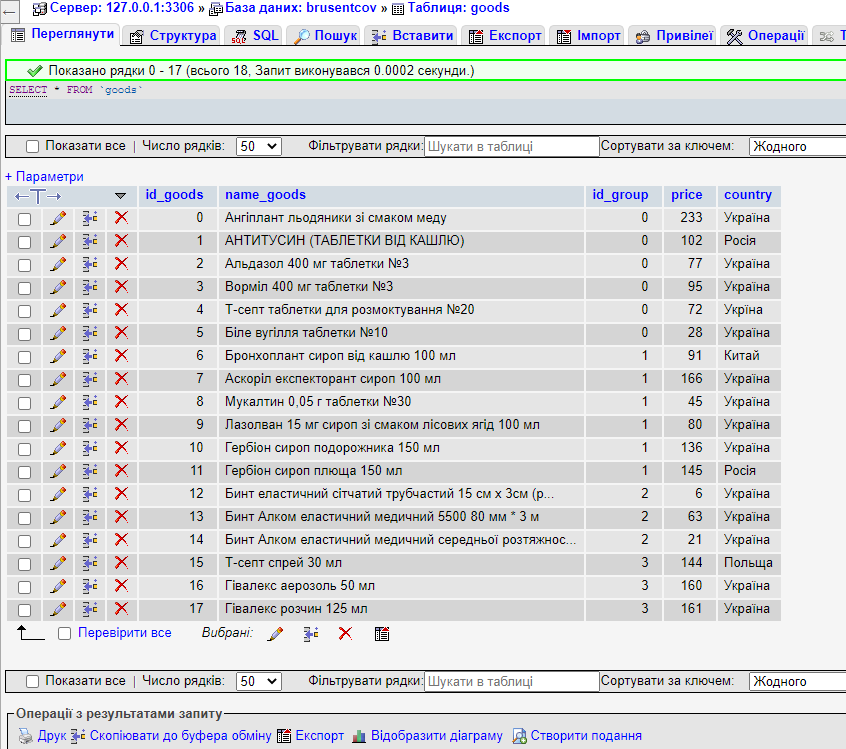
Для редагування та перегляду даних в базі даних використаємо phpMyAdmin. Для цього потрібно обрати базу, після цього таблицю, в якій потрібно змінити данні, а далі натиснути на вкладку «Вставити»(рисунок 4.13).  
  


Рисунок 4.13 – Пергляд даних в phpMyAdmin

Після вводу потрібних даних натискаємо на кнопку «Виконти»(рисунок 4.14). І наші данні записуються в базі.

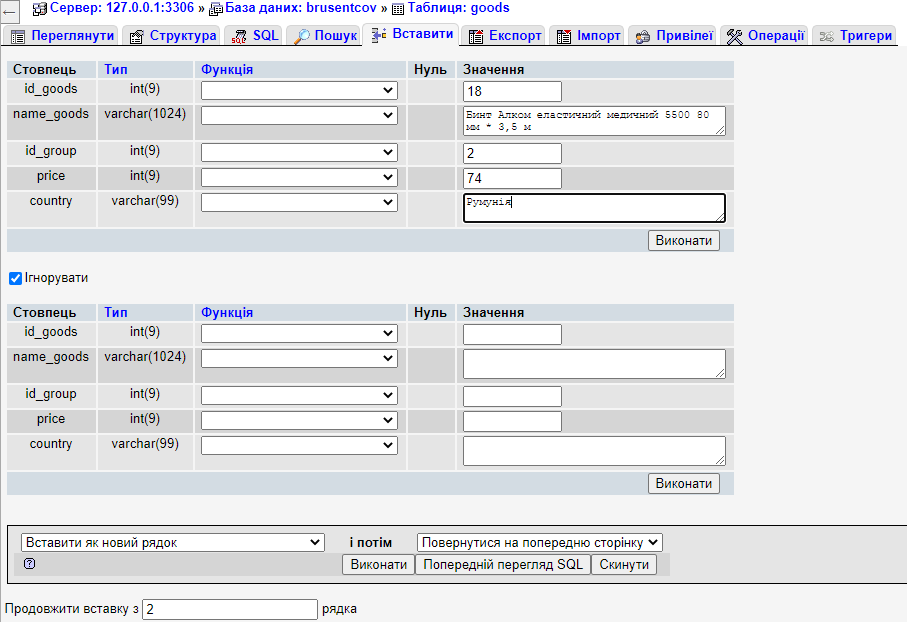


Рисунок 4.14 – Редагування даних в phpMyAdmin

## 4.2 Опис архітектури програмної системи

UML – діаграма програмної системи(рисунок 4.13). UML Це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю.

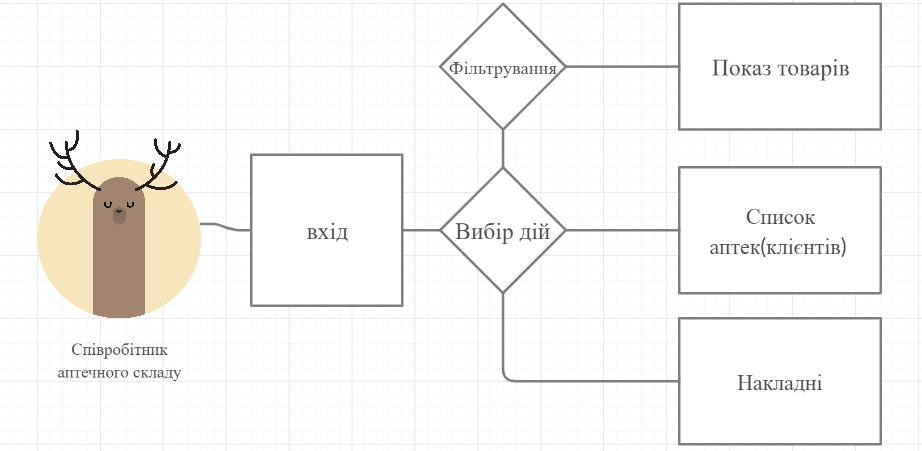


Рисунок 4.13 – Діаграма програмної системи

Кожен користувач проходить авторизацію за допомогою системи «Вхід».

Системою користується тільки співробітник складу. Робітник отримує доступ до читання, створення та редагування даних. Має можливість налаштувати базу даних через таблиці, які оновлюють данні на сервері.

## 4.3 Опис концептуальної моделі бази даних

На сьогоднішній день phpMyAdmin широко застосовується на практиці. Останнє пов'язано з тим, що розробники інтенсивно розвивають свій продукт, з огляду на всі нововведення СКБД MySQL. Переважна більшість українських провайдерів використовують цей застосунок як панель керування для того, щоб надати своїм клієнтам можливість адміністрування виділених їм баз даних.

Програма розповсюджується під ліцензією GNU General Public License і тому деякі інші розробники інтегрують його у свої розробки, наприклад XAMPP, Denwer.

На (рисунку 4.14) потрібно авторизуватись через phpMyAdmin, для редагування та читання бази даних. Логін для входу «root», пароль «» - пустий.

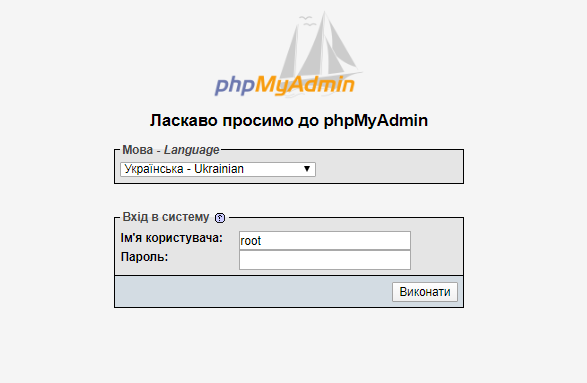


Рисунок 4.14 – Авторизація в phpMyAdmin

Тут ми має змогу переглянути список баз даних (рисунок 4.15). При натисненні на посилання будь-якої бази даних ми можемо переглянути вміст бази на таблиці (рисунок 4.16), ми відкрили базу даних «brusentcov», в яку і будемо зв’язувати з програмою написаною на мові c#.

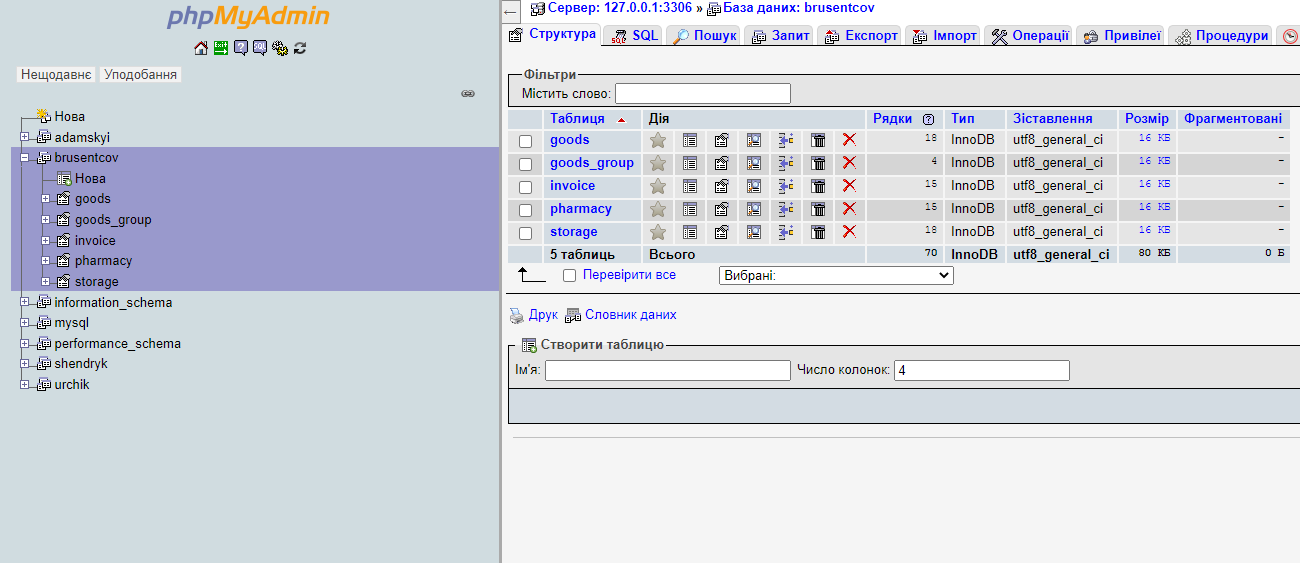


Рисунок 4.15 – Список баз даних

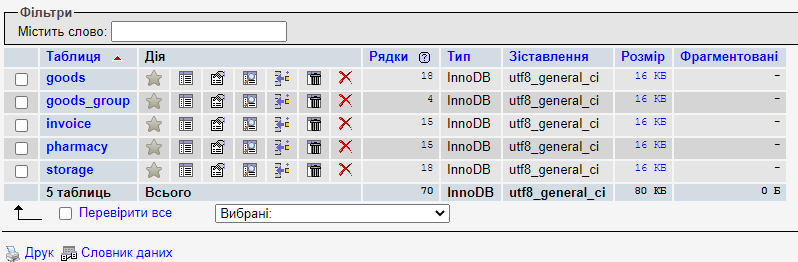


Рисунок 4.16 – Вміст бази даних «brusentcov»

Заповнювати базу даних можна за допомогою консолі, або спеціальної вкладки «Вставить». Найкращий варіант заповнення бази, для мене – це спосіб через вкладку «Вставить»(рисунок 4.17). Для цього потрібно обрати таблицю, наприклад «goods». А потім натиснути кнопку «Вставить», і додати потрібну інформацію, далі натиснути кнопку «Далі», щоб записати запис в базі.

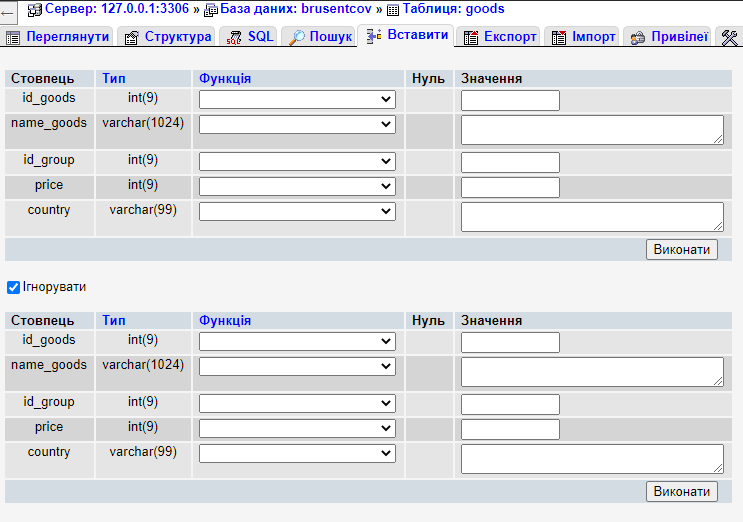


Рисунок 4.17 – Додавання інформації в таблицю

Для створення нової таблиці, потрібно натиснути «створити нову базу». Після цього вписати потрібні типи рядків в таблиці, та присвоїти їм індивідуальні назви. При цьому надавши додаткову характеристику, якщо це потрібно.

На рисунку 4.18 зображено структуру таблиці товарів - goods. Обмеження таблиці id\_goods, name\_goods, id\_group та інші атрибути не можуть дорівнювати NULL.

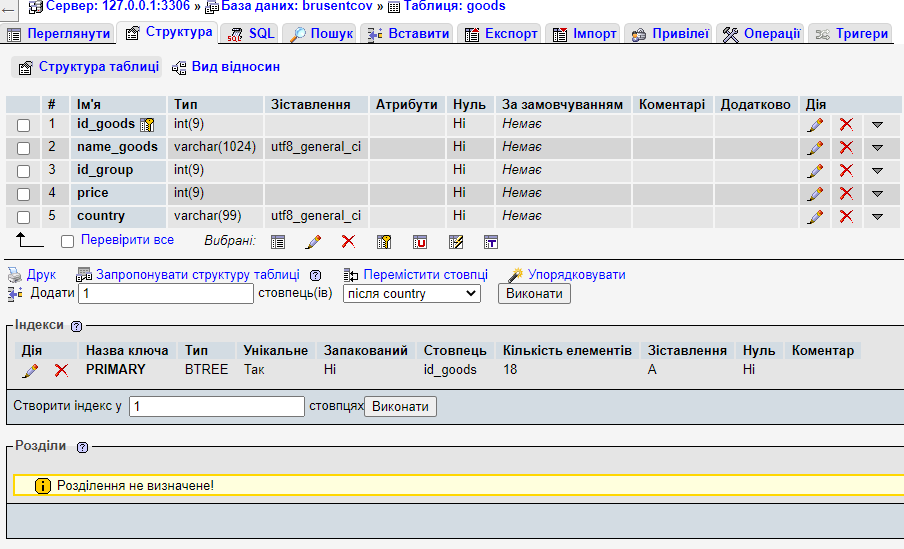


Рисунок 4.18 – Структура таблиці товарів

На рисунку 4.19 зображено структуру таблиці групи товарів – goods\_group. Має в собі групу для кожного товару.

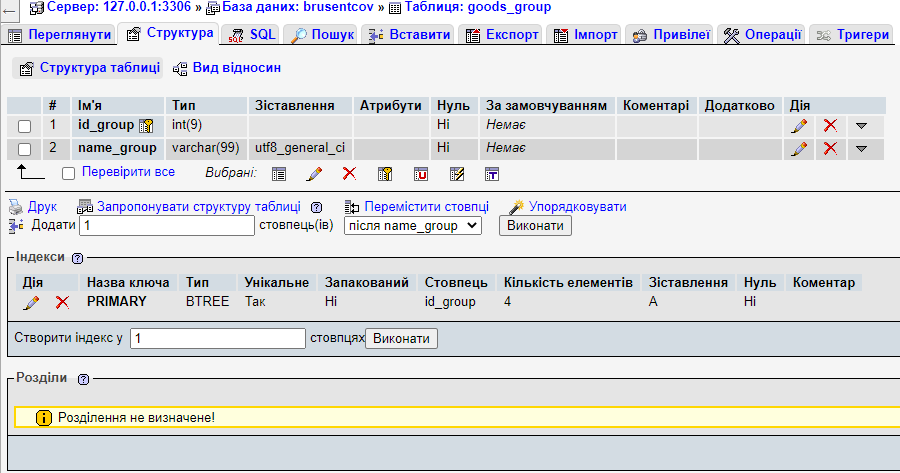


Рисунок 4.19 – Структура таблиці груп товарів

На рисунку 4.20 зображено структуру таблиці накладних – invoice. Має в собі дані клієнта(аптеку), який замовив доставку товару, та дату виїзду.

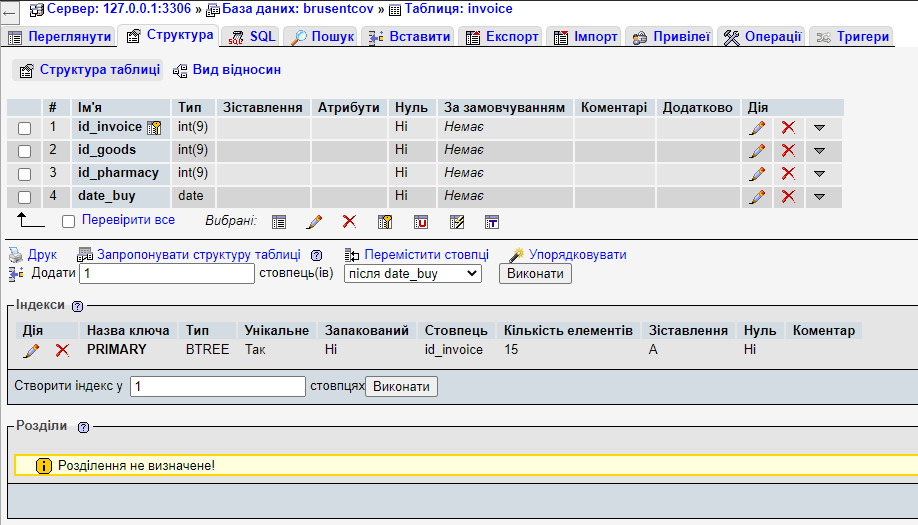


Рисунок 4.20 – Структура таблиці накладних

На рисунку 4.21 зображено структуру таблиці аптек – pharmacy. Це список аптек, які замовляють доставку.

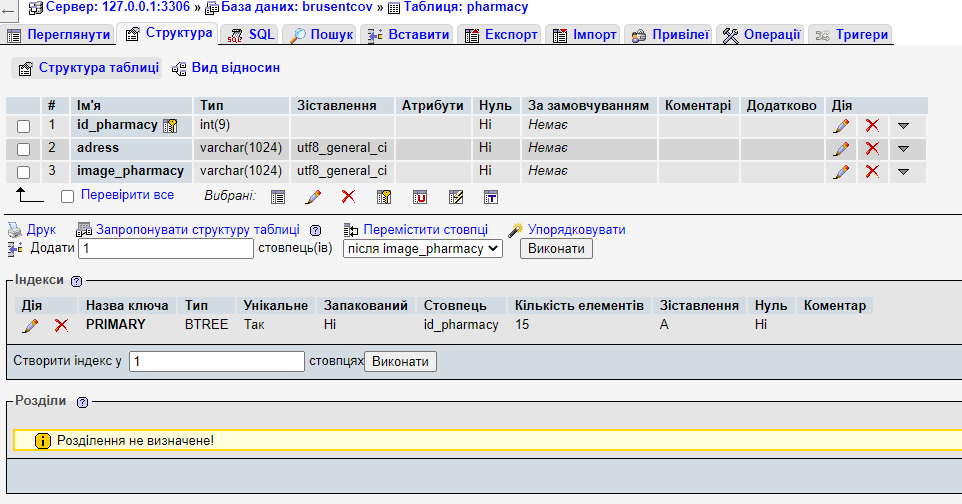


Рисунок 4.21 – Структура таблиці аптек

На рисунку 4.22 зображено структуру таблиці сховища – storage. Це місце, де знаходиться інформація про всі товари та дату прибуття на склад.

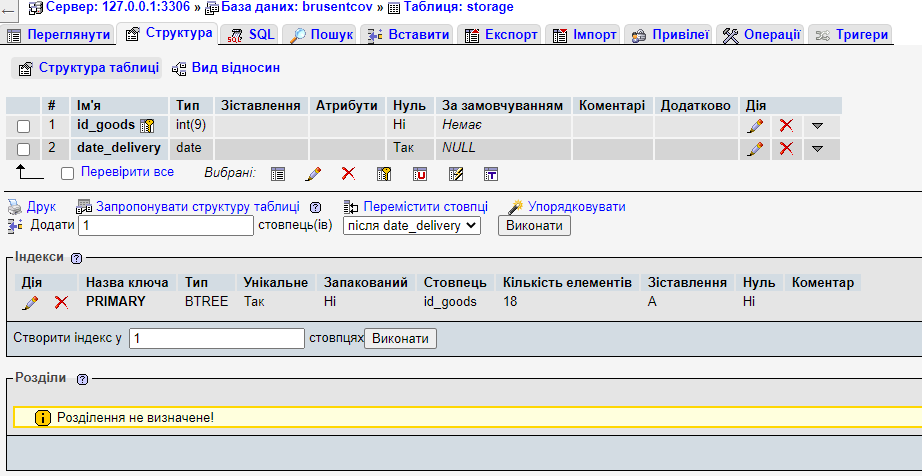


Рисунок 4.22 – Структура таблиці сховища

## 4.4 Опис програної реалізації

Підключення бази даних до візуального додатка написаного на мові С# відбувається за допомогою спеціального класу connctingToBD. В якому прописані всі налаштування для підключення до бази (рисунок 4.23).



Рисунок 4.23 – Підключення до бази через клас connectionToBD

Для підключення до бази нам потрібно ввести пароль та логін до бази, та вказати назву сервера, а також назву бази даних. Пароль: «», логін: «root», сервер: «localhost», база даних: «brusentcov».

## 4.5 Опис задач автоматизації та інтерфейсу користувача

Після запуску програми користувач потрапляє на головну форму авторизації (рисунок 4.24), де потрібно обрати наступну дію: продовжити.

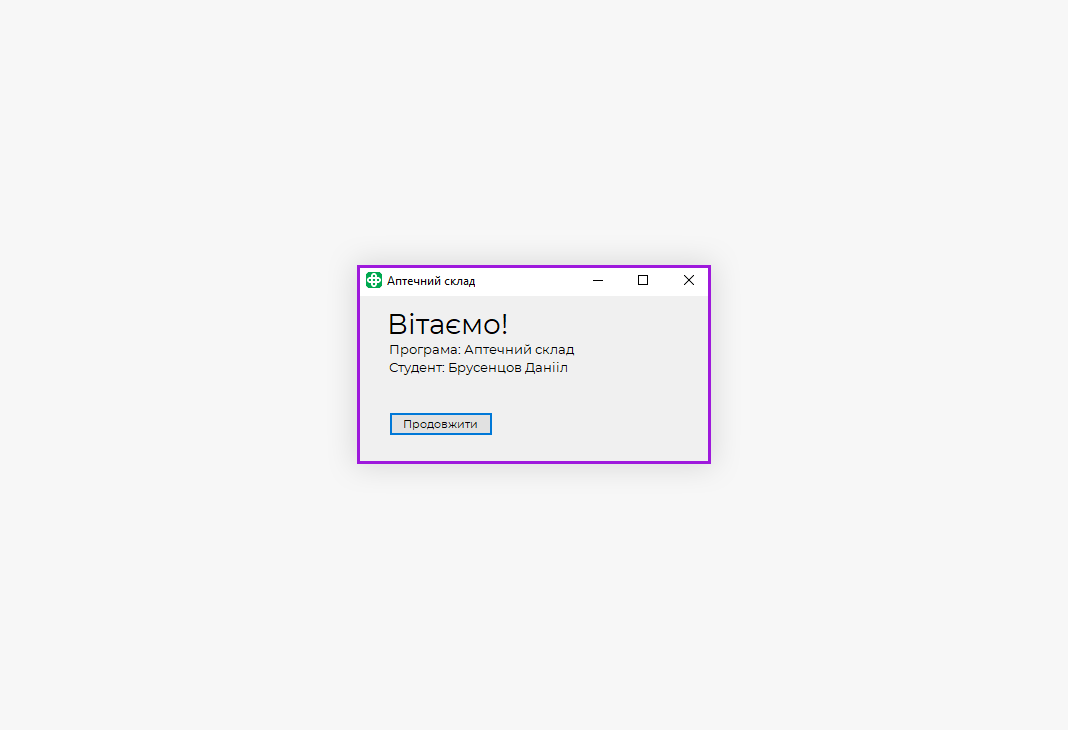


Рисунок 4.24 – Початкова форма

Користувач, а саме співробітник аптечного складу має змогу користуватися всім функціоналом системи. З’являється головне вікно(рисунок 4.25), тут можна моніторити данні про товари.

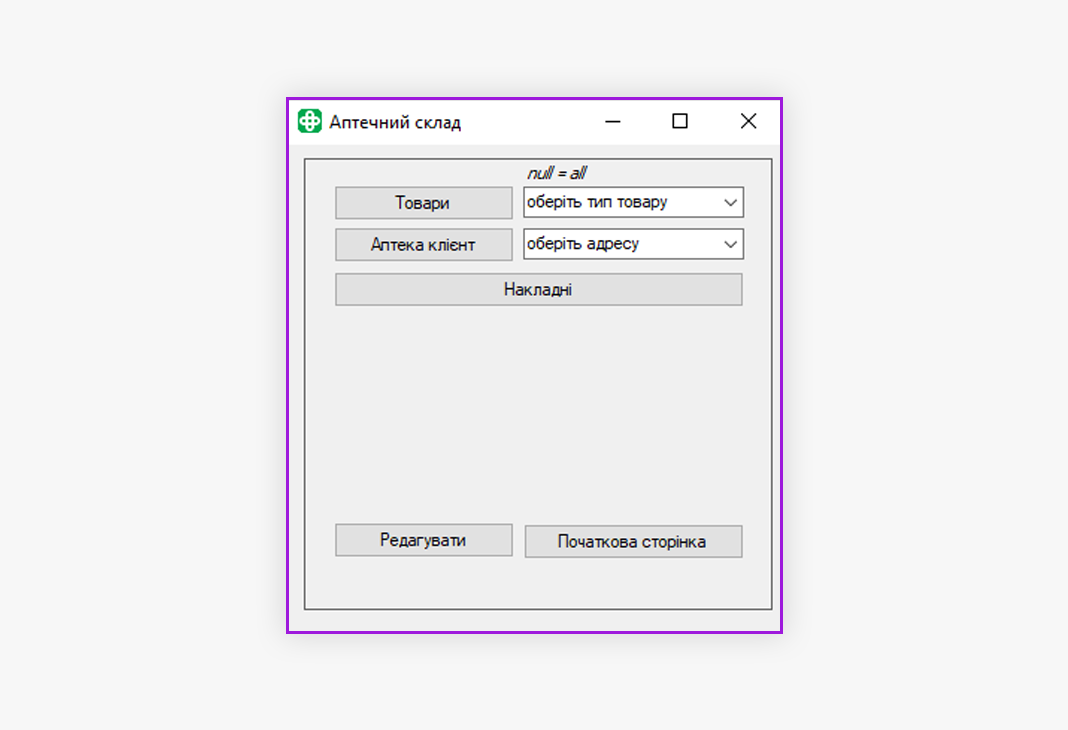


Рисунок 4.25 – Форма головного меню

Тут користувач може обрати дії зв’язані з пошуком товарів на складі (рисунок 4.26), звичайно, використавши фільтр по групам. На даний момент є чотири групи товарів на складі, а саме: таблетки, сироп, бинти, спрей. Кожен товар має свою групу, завдяки цьому фільтрування відбувається дуже швидко.

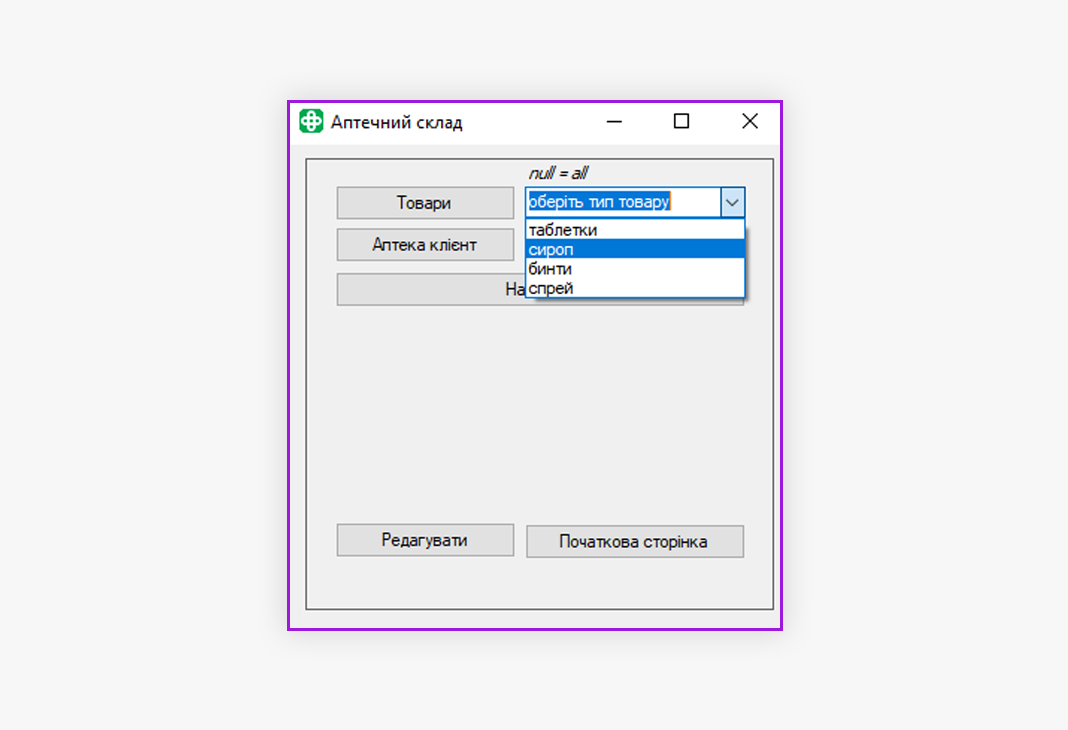


Рисунок 4.26 – Форма фільтрація товарів

Якщо обрати групу товарів «таблетки» ми побачимо відфільтрований список товарів, які відповідають цій групі (рисунок 4.26), якщо ж співробітник нічого не вибере, то він побачить перелік всіх товарів, які зберігаються на складі (рисунок 4.27).

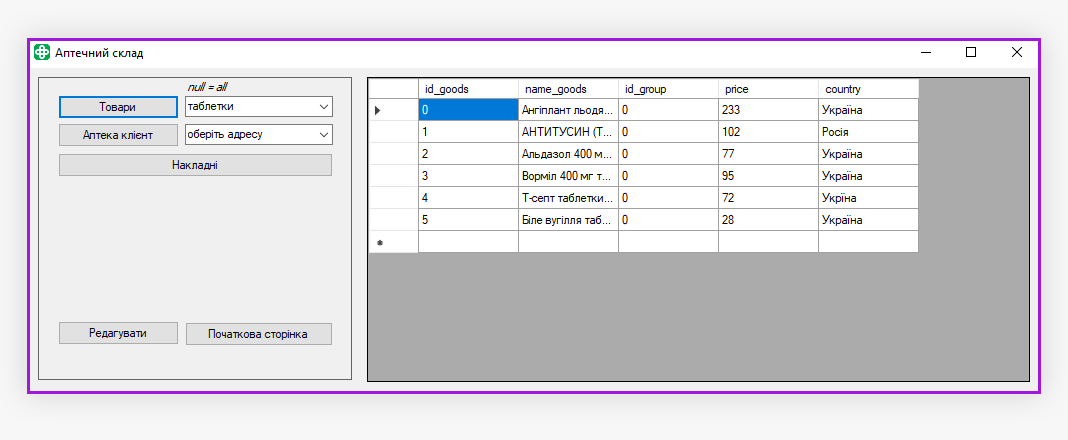


Рисунок 4.26 – Профіль фільтр товарів «таблетки»

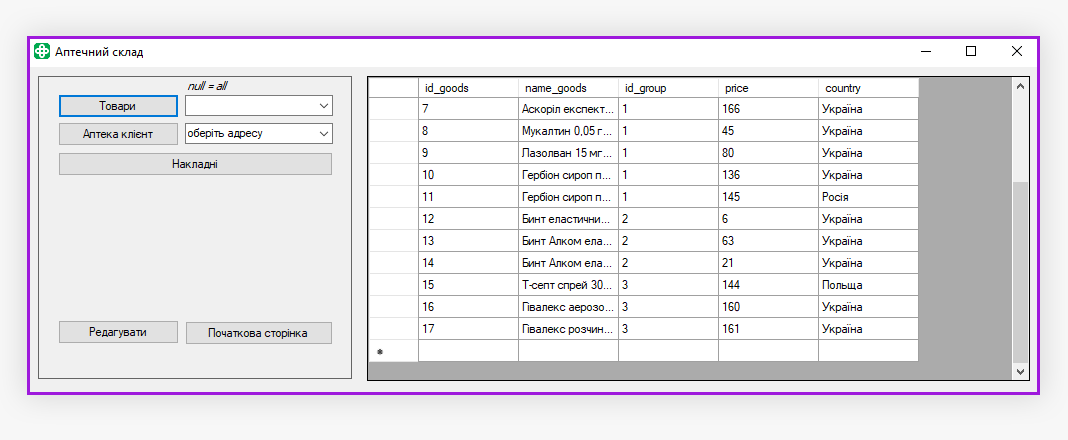


Рисунок 4.27 – Профіль фільтр товарів «null»

На цьому співробітник може перейти до моніторингу аптек – клієнтів, які використовують послуги складського приміщення, якщо не обирати фільтрування, то він побачить список всіх клієнтів(рисунок 4.28).

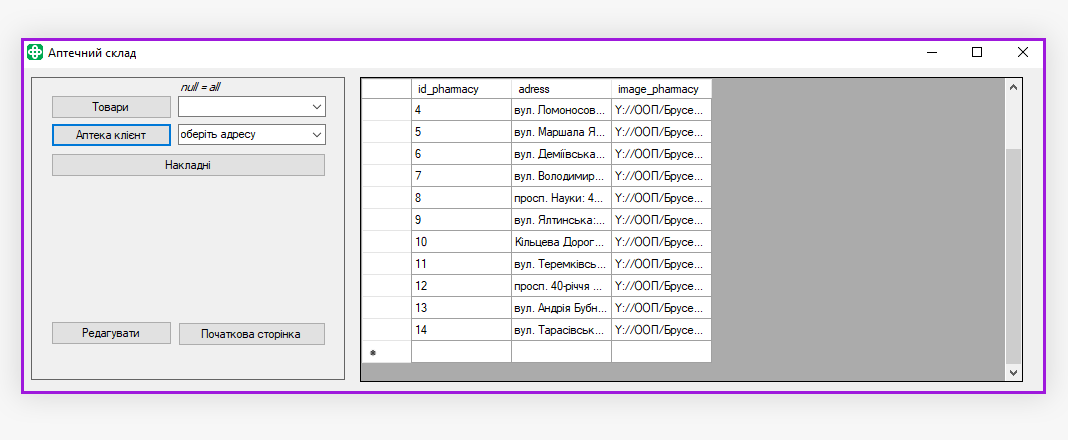


Рисунок 4.28 – Перегляд списку клієнтів

Якщо він вибере по фільтру окрему адресу аптеки, він отримає інформацію про неї, та побачить фото аптеки(рисунок 4.29).

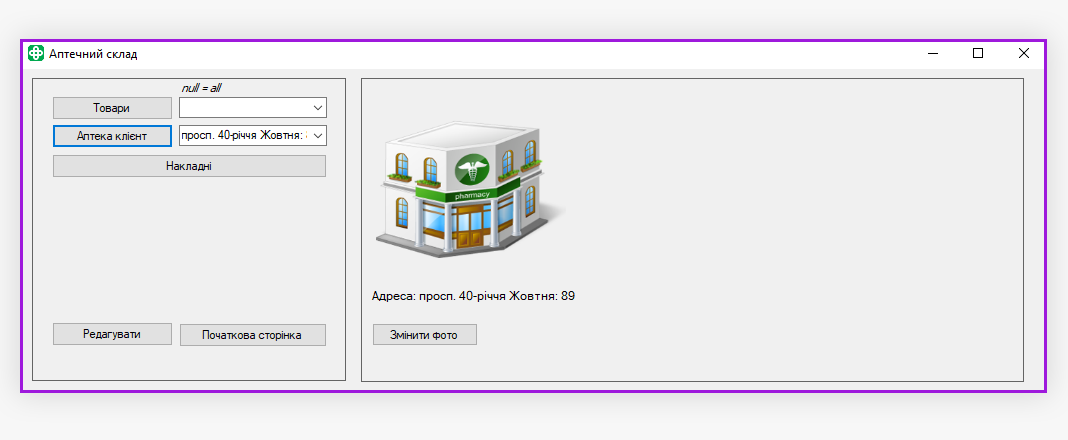


Рисунок 4.29 – Перегляд інформації про аптеку

Тут можна змінити фотографію аптеки, вона автоматично завантажиться на сервер та завантажиться після наступного входу в систему (рисунок 4.30). Також можна переглянути данні про накладні, натиснувши на кнопку - накладні(рисунок 4.31).

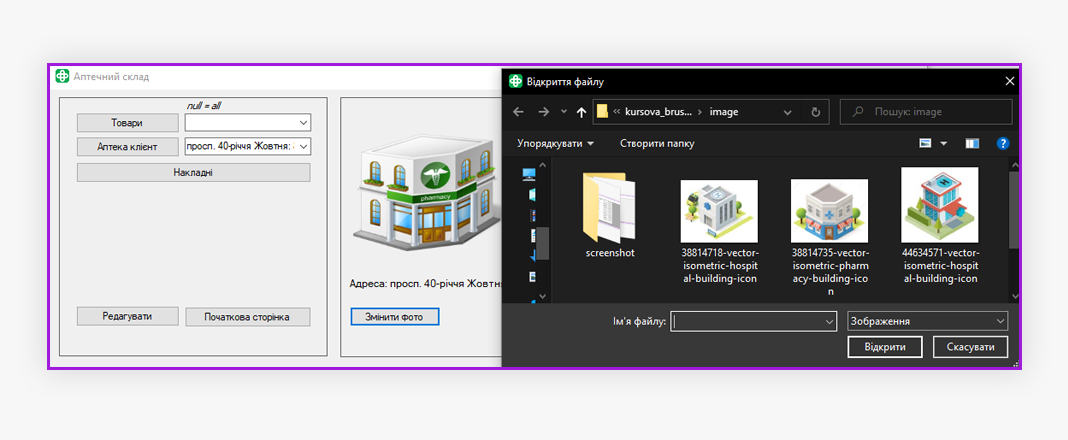


Рисунок 4.30 – Зміна ілюстрації аптеки

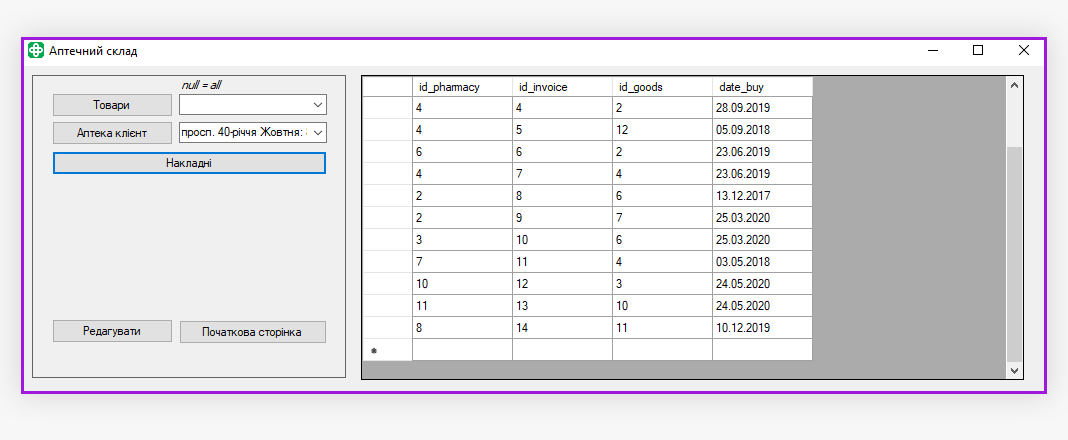


Рисунок 4.31 – Список накладних

В програмі, звичайно ми можемо редагувати данні. Для цього потрібно обрати будь-яку таблицю та два рази натиснути на комірку, в якій потрібно зробити зміну. Після редагування, просто потрібно натиснути на кнопку – редагувати. Тоді данні збережуться на сервері, якщо не було допущено помилок, і будуть завантажуватись в систему з БД(рисунок 4.32).

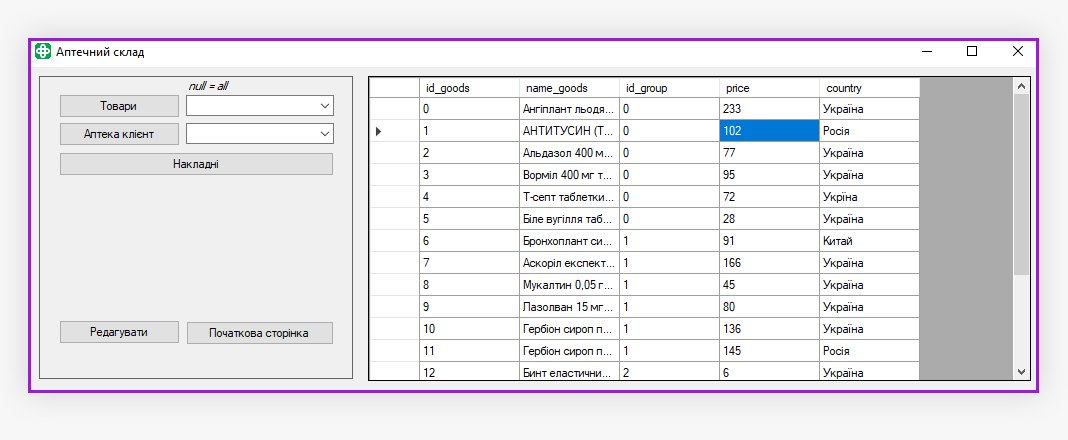


Рисунок 4.32 – Редагування даних

# ВИСНОВКИ

Під час виконання курсової роботи було вдосконалено навички з програмування на мові С# та SQL, повторено основні конструкції мови, властивості базових компонентів та впровадив з’єднання візуальних компонентів з сервером за допомогою SQL запитів. Під час створення додатків були використані наступні компоненти: textBox, label, richTextBox, dataGridView, button.

Команди SQL можуть виконуватись як над цілою групою таблиць, так і над одним об’єктом. Завдяки використанню SQL можливо створювати бази даних, виконувати над ними дії та робити запити до баз даних. Частіше SQL використовують для запитів. Запит - це команда, яка формується для однієї чи декількох таблиць бази даних і виводить певну інформацію з них. Запити не змінюють інформацію у таблицях, а лише показують її користувачу. Усі запити SQL формуються на базі однієї нескладної за структурою команди - SELECT. У найпростішому вигляді команда SELECT дає інструкцію базі даних для пошуку інформації у таблиці.

База даних розроблялася у додатку phpMyAdmin. phpMyAdmin - веб-додаток з відкритим кодом на мові PHP із графічним веб-інтерфейсом для адміністрування бази даних MySQL або MariaDB. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних.

За візуальну сторону даного проекту відповідав додаток створений в Visual Studio. Яким вдалось вдало оволодів під час курсу з ООП.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Радченко С.П. До питання про інформатизацію самостійної роботи студента-математика. Міжнародна науково-практична конференція «Проблеми та перспективи фахової підготовки вчителя математики», Вінницький державний педагогічний університет імені Михайла Коцюбинського. – Вінниця, 2012
2. Коновалов Я.Ю., Соболев С.К., Ермолаева М.А. Методические аспекты автоматической генерации задач по линейной алгебре // Инженерный журнал: наука и инновации. 2013. вып. 5. 14 с.
3. Радченко С. П. Використання методу шаблонів при формуванні самостійних завдань для студентів з курсу лінійної алгебри, Неперервна професійна освіта: теорія і практика (1-2), 2016, с. 85- 90. ISSN 1609-8595
4. Ульман Дж.Д. Основы систем баз данных. – М.: Финансы и статистика, 1983. – 334 с.
5. Мартин Грабер. SQL. – К.: Изд-во “Лори”, 2003. – 644 с.
6. Аткинсон, Леон. MySQL. Библиотека профессионала.: Пер. с англ. – М.: Изд. дом &quot;Вильямс&quot;, 2002. — 624 с.
7. Гарсия-Молина, Гектор, Ульман, Джеффри, Уидом, Дженнифер. Системы баз данных. Полный курс.: Пер. с англ. – М.:Изд. дом “Вильямс”, 2003. – 1088 с.
8. Теория и практика построения баз данных. 8-е изд. / Д.Кренке. – СПб.: Питер, 2003. – 800 с.
9. Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. – М.: Финансы и статистика, 1989.
10. Глушаков С.В., Ломотько Д. В. Базы данных. (Учебный курс). – Харьков, 2002. – 504 с.
11. https://professorweb.ru/
12. https://metanit.com/sharp/
13. http://www.cyberforum.ru/
14. https://devcolibri.com/sql/
15. https://stackoverflow.com/sql
16. https://www.khanacademy.org/computing/computer-programming/sql
17. https://www.udacity.com/course/intro-to-relational-databases--ud197
18. https://academy.vertabelo.com/
19. https://mode.com/sql-tutorial/introduction-to-sql/
20. http://www.sql-ex.ru/
21. https://www.add.ua/ua/medicinskie-izdelija/perevjazochnye-sredstva/binty/binty-jelastichnye/