## ЗВІТ

Запити:

1. Багатотабличний з групуванням(**параметричний**)

*Умова:* Вивести інформацію про кількість проданих товарів категорії,що задана параметром, відсортованих за спаданням(**CategoryName** як параметр)

***SQL-код запиту****:*

SELECT S.product\_name, S.promotional\_product, SUM(product\_number) AS sold\_amount

FROM Sale

INNER JOIN(SELECT P.product\_name, P.characteristics, UPC, SP.promotional\_product

FROM Store\_Product SP

INNER JOIN(SELECT \*

FROM Product

WHERE category\_number IN (SELECT category\_number

FROM Category

WHERE category\_name=’**CategoryName**’

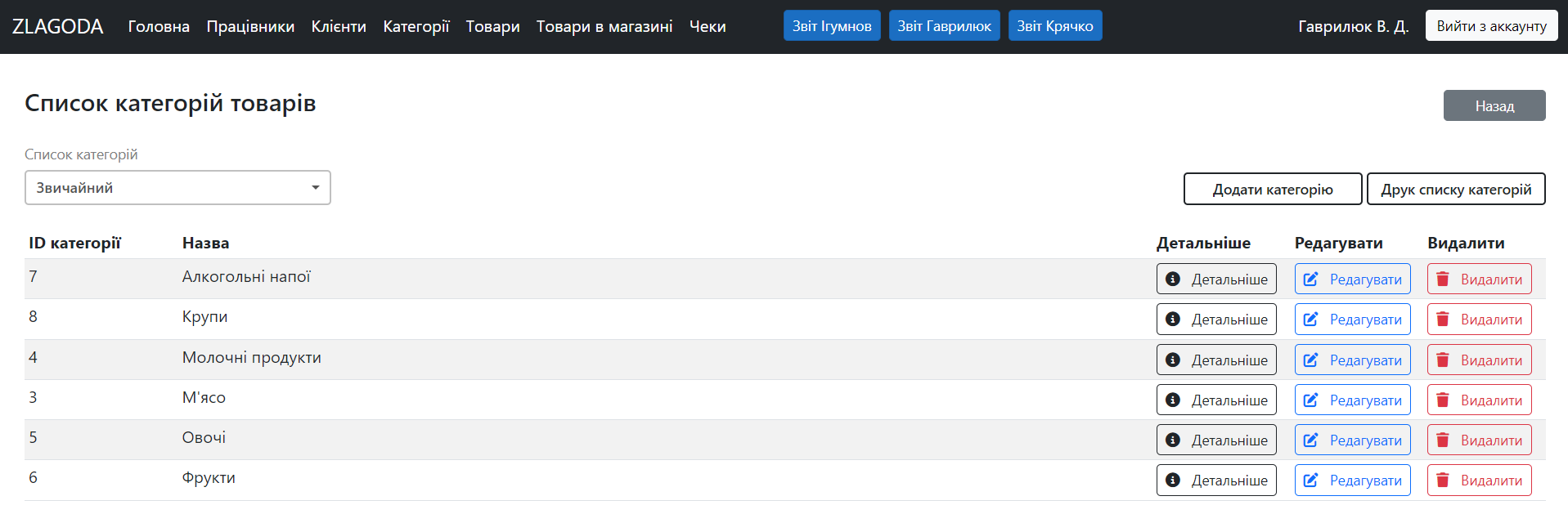
)

) AS P ON SP.id\_product=P.id\_product

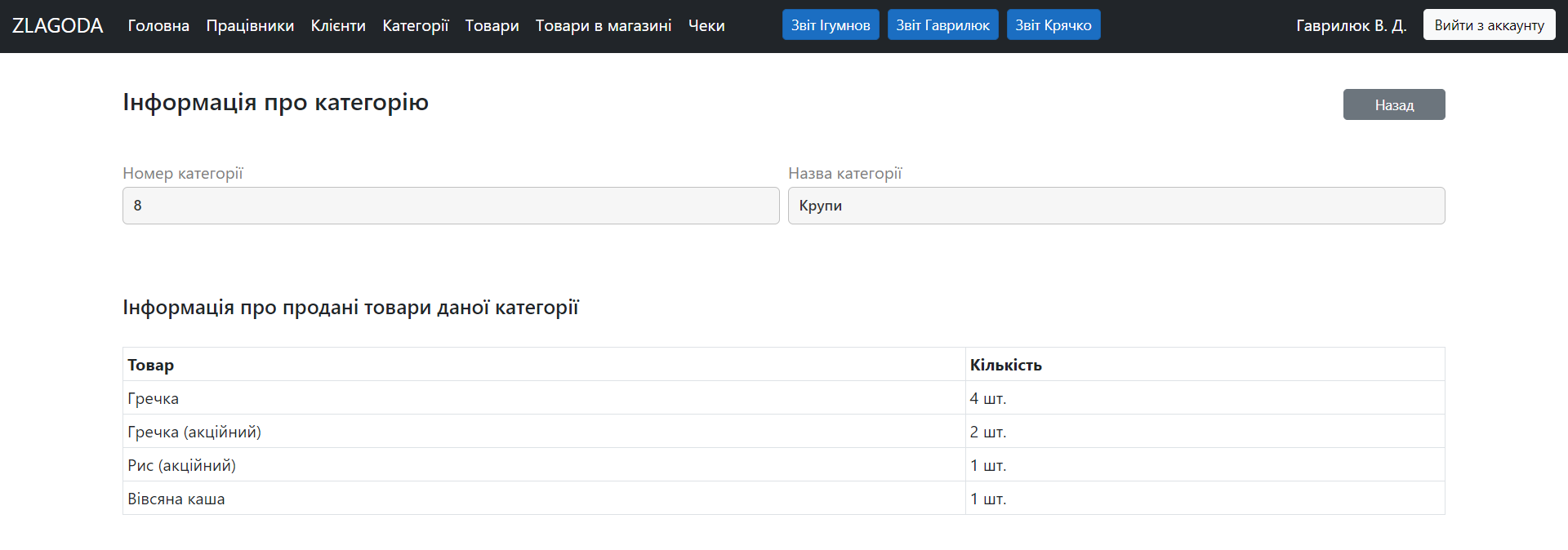
) AS S ON Sale.UPC=S.UPC

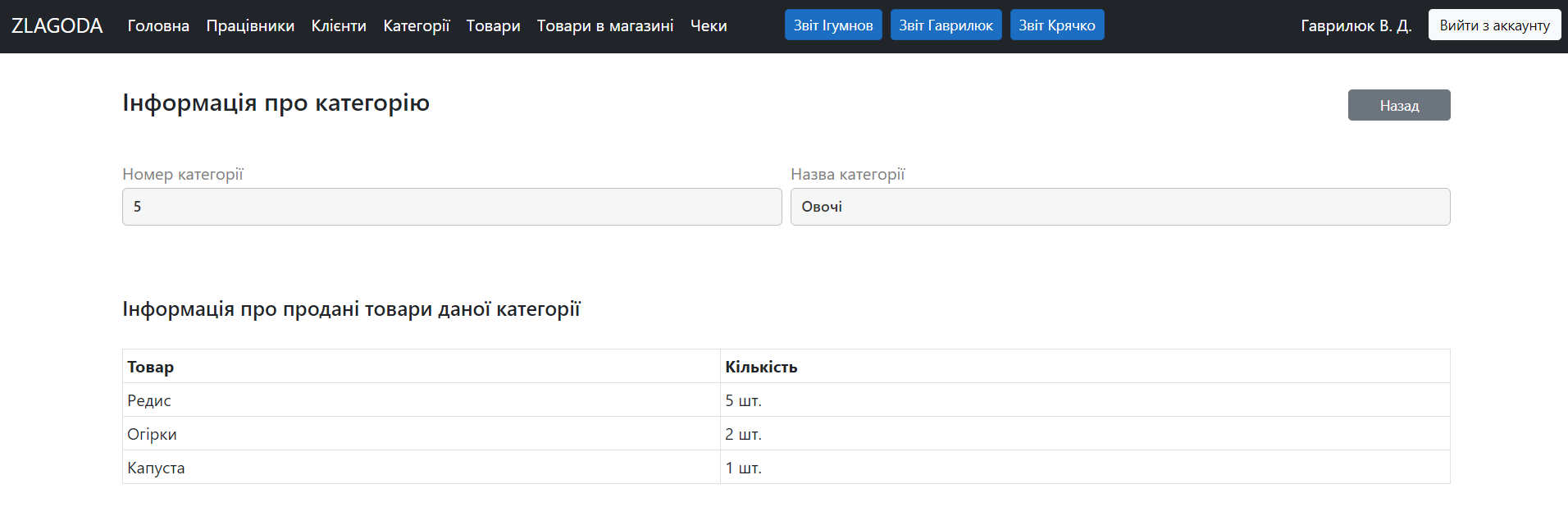
GROUP BY S.product\_name, S.promotional\_product

ORDER BY sold\_amount DESC

*Скріншот з інтерфейсу користувача(Список всіх наявних категорій ,при натисканні на кнопку детальніше виводиться результат запиту):*

*Результати запиту:*

**Перший результат:

Другий результат:

*Виводяться всі товари заданої категорії,що були продані(містяться в чеку) і яку їх кількість було продано.*

1. Багатотабличний з подвійним запереченням(**параметричний**)

*Умова:* Знайти касирів,які обслуговували всіх покупців касира ,прізвище якого вказується параметром(як параметр в запиті вказується **EmplSurname**)

***SQL-код запиту****:*

SELECT empl\_surname, empl\_name, empl\_patronymic

FROM Employee Emp

WHERE NOT EXISTS(SELECT card\_number,E.id\_employee

FROM [Check]

INNER JOIN Employee E

ON E.id\_employee=[Check].id\_employee

WHERE E.empl\_surname=’**EmplSurname**’

AND card\_number NOT IN(SELECT card\_number

FROM [Check]

WHERE Emp.id\_employee=id\_employee

)

)

AND EXISTS(SELECT \*

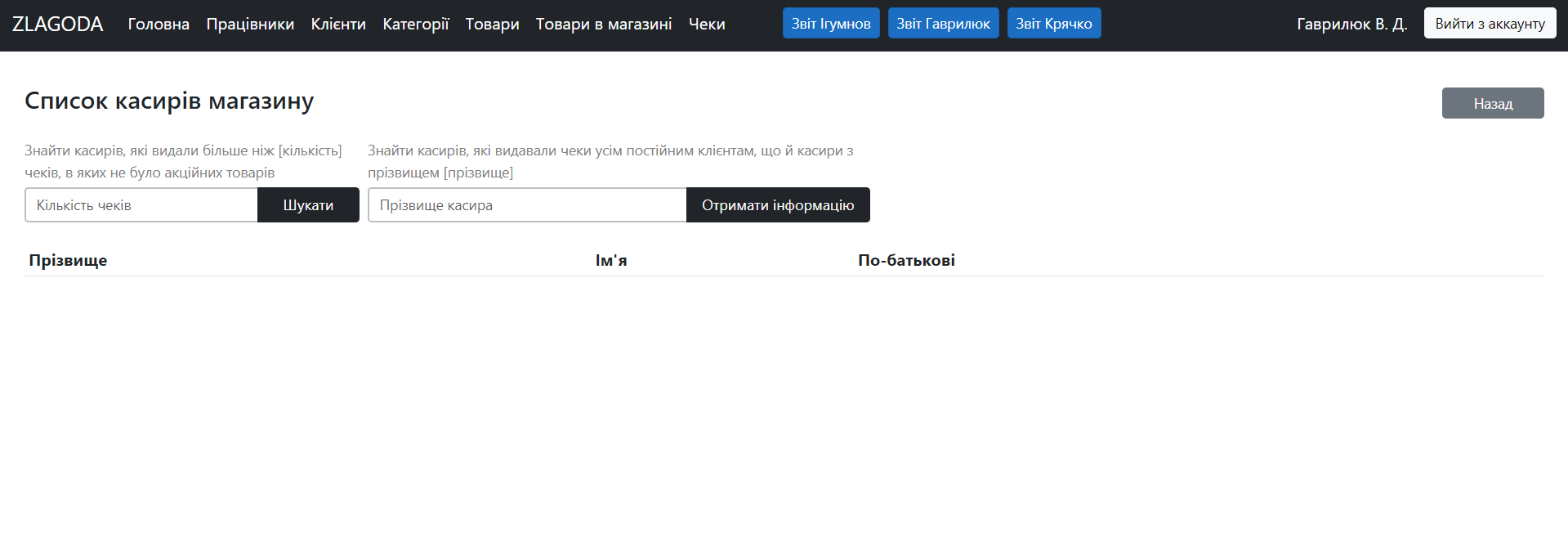
FROM [Check]

INNER JOIN Employee

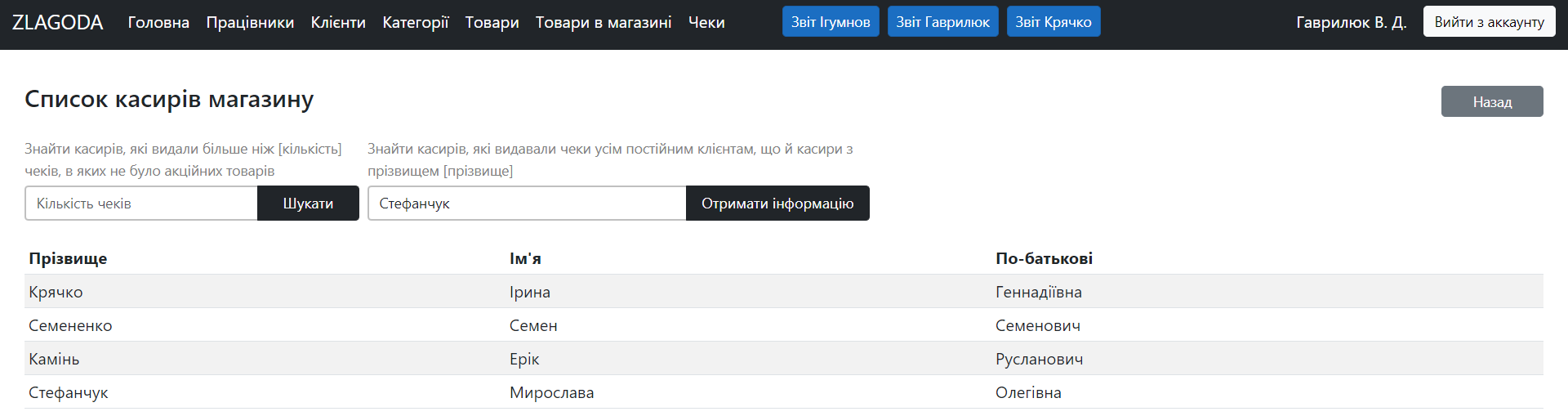
ON [Check].id\_employee=Employee.id\_employee

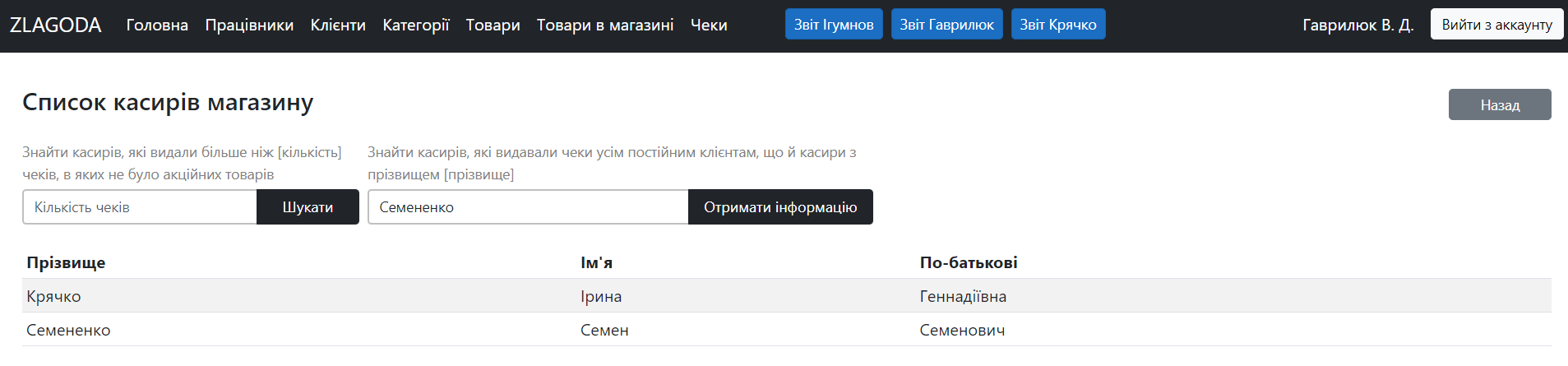
WHERE Employee.empl\_surname=’**EmplSurname**’

)

*Скріншот з інтерфейсу користувача(обрана додаткова опція для пошуку касирів):*

*Результати запиту(отримані при введені прізвища і натисканні на кнопку «Отримати інформаціию»):*

Перший результат:

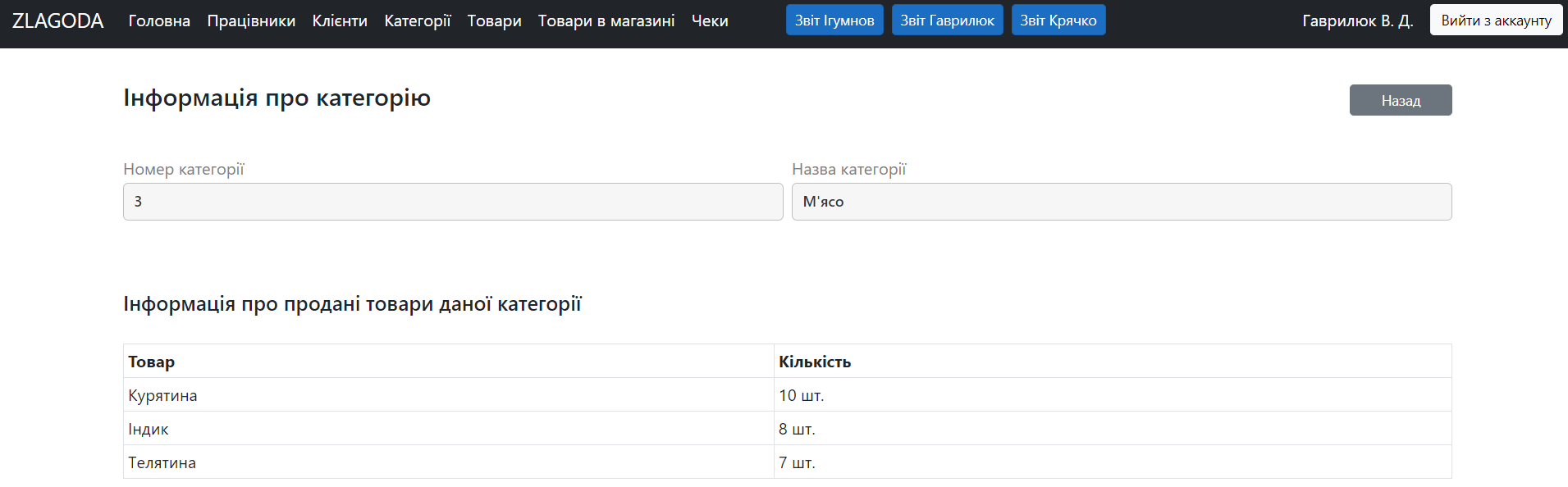
Другий результат:

При введенні прізвища( в першому прикладі «Стефанчук») ми отримуємо список імен всіх касирів,які обслуговували всіх клієнтів касира,прізвище котрого ми ввели(тобто,Стефанчук)

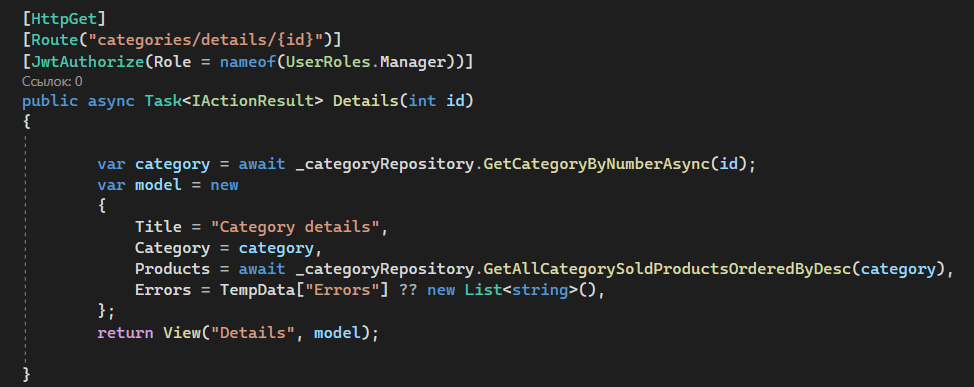
**Демонстрація коду прикладної програми**

Демонстрація коду на прикладі першого запиту:

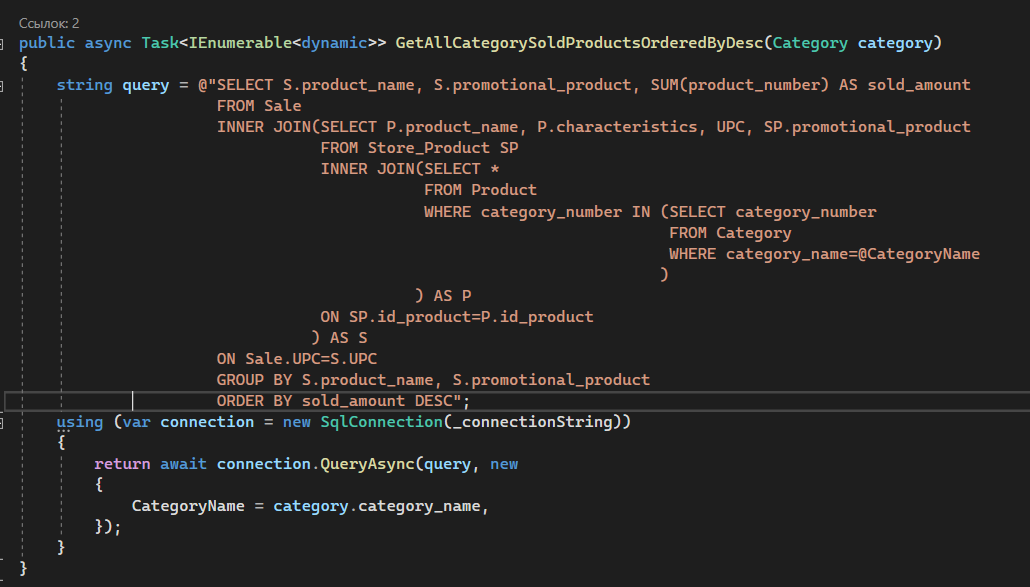
1. При переході з сторінки категорій на деталі категорій до URL-адреси додаєтьсяза яким номером категорії ми хочемо оглянути її деталі(а точніше кількість проданого товару).

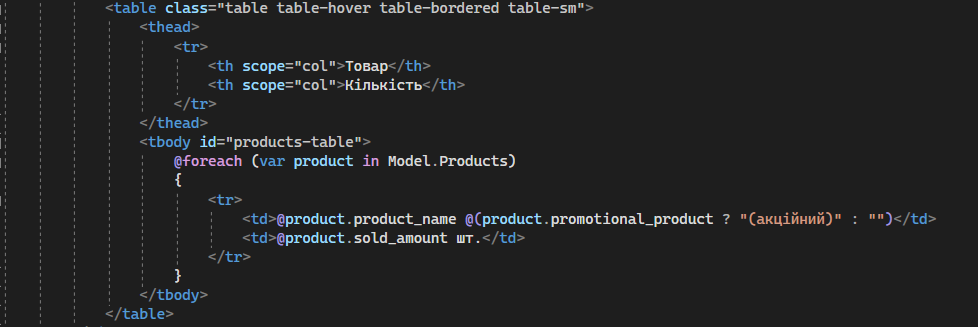


1. При переході на сторінку деталей категорії ми повинні повернути контролер з результатом запиту.Перед методом є декілька атрибутів,атрибут “JwtAuthorize” вказує на те,що цей метод доступний лише для ролі Менеджера,атрибут “Route” демонструє,за яким маршрутом знаходиться ця сторінка.Врешті-решт,метод повертає View,тобто HTML-сторінку,котра містить об’єкт ,що в свою чергу містить результат методу *GetAllCategorySoldProductsOrderedByDesc* ,який реалізує запит до бази даних і повертає необхідний нам результат



1. Власне сама реалізація методу *GetAllCategorySoldProductsOrderedByDesc* ,як бачимо,як аргумент приймає категорію,в змінній query вказаний текст SQL-запиту,необхідного для подальшого виконання.Далі встановлюємо з’єднання з БД через змінну *\_connectionString* і виконуємо наш запит через метод QueryAsync(асинхронний запит) об’єкту *connection*.



1. Повернувши через контролер об’єкт з результатом запиту та іншими даними,ми виводимо його в HTML-сторінці у вигляді таблиці засобами мови С#, в циклі *foreach* йдучи по списку всіх товарів заданої категорії та кількості проданих одиниць(*sold\_amount*),також є перевірка на акційність товару(*promotional\_product*)

Зрештою,користувач отримує результат на екрані,в якому демонструється таблиця,в котрій містяться товари певної категорії і кількість раз,коли їх купили.