# Звіт

Запити:

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

Умова: Вивести інформацію про касирів, які видали більше заданої параметром кількості чеків,в які не входили акційні товари.(параметр **amount**)

***SQL-код запиту****:*

SELECT E.empl\_surname, E.empl\_name, E.empl\_patronymic

FROM Employee E

WHERE EXISTS (SELECT check\_number, COUNT(check\_number) AS check\_amount

FROM [Check] H

GROUP BY check\_number

HAVING (SELECT COUNT(check\_number) as check\_amount

FROM [Check] ССС

WHERE id\_employee=E.id\_employee

AND NOT EXISTS(SELECT UPC

FROM Sale

WHERE check\_number=ССС.check\_number

AND UPC IN(SELECT UPC

FROM Store\_Product

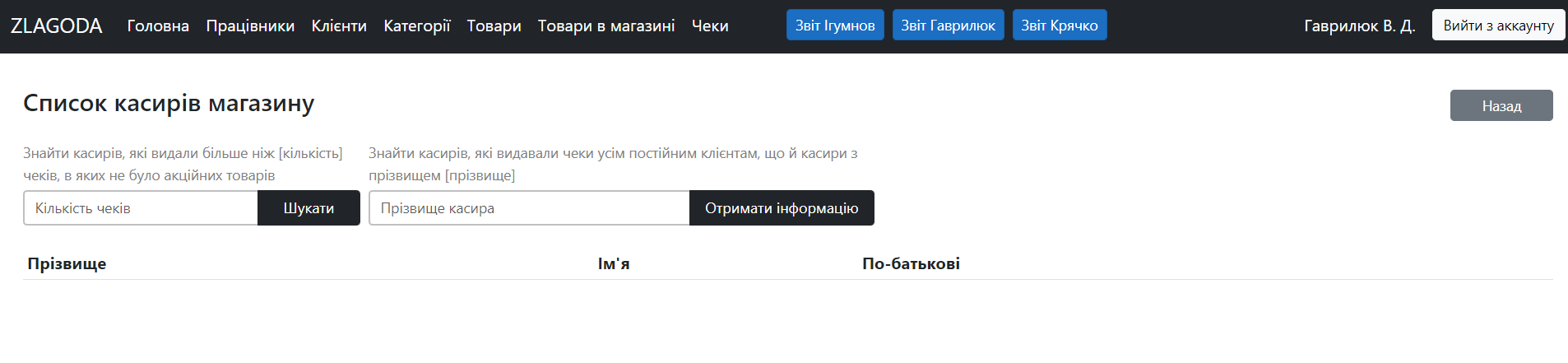
WHERE promotional\_product=1

)

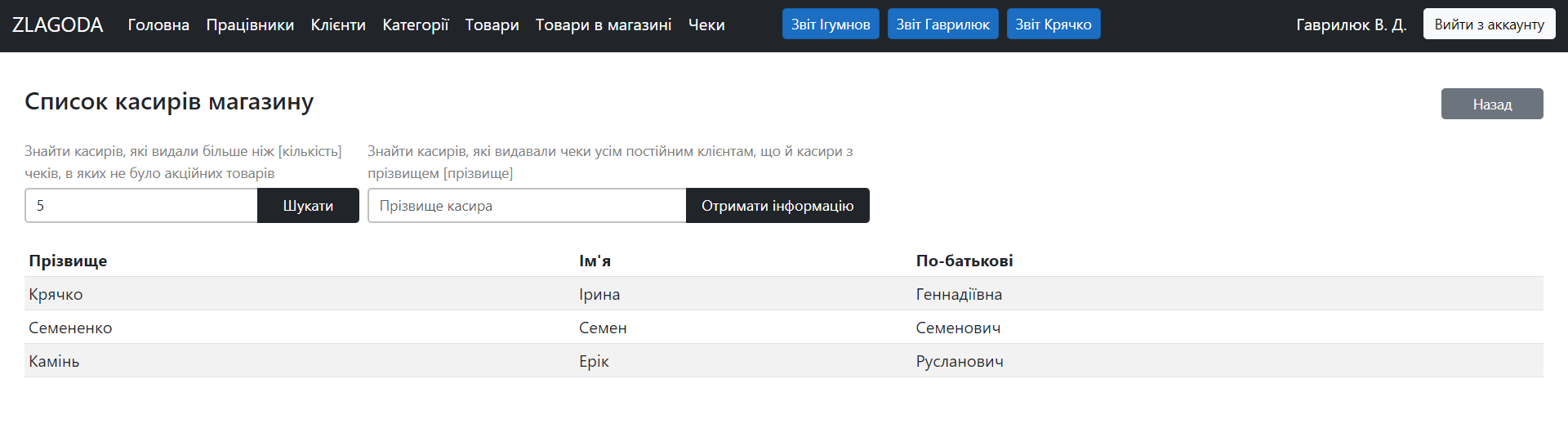
)

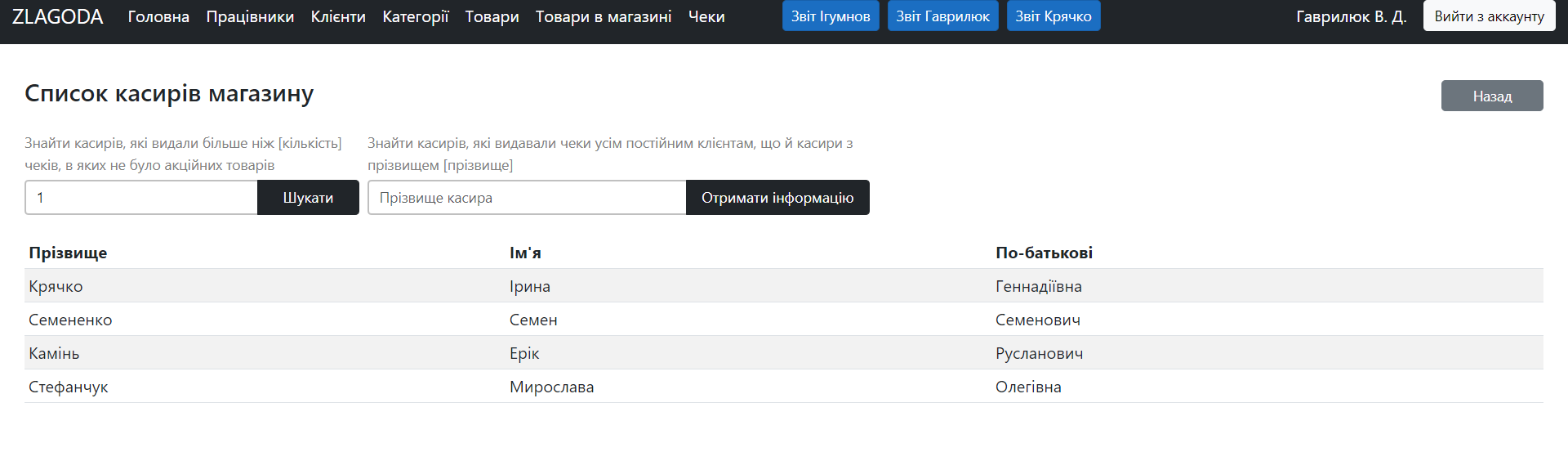
)> **amount**

)

*Скріншот з інтерфейсу користувача(в полі кількість чеку вводимо число):*

*Результат запиту(список всіх касирів,що підходять умові):*

Перший результат

Другий результат

Виводиться ім’я всіх касирів,що видали більше чеків заданого параметром значення ,в які не входили акційні товари.

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

Умова: Визначити покупців,які купили всі товари в магазині,що і покупець ,прізвище якого задано параметром **CustSurname**

***SQL-код запиту****:*

SELECT \*

FROM Customer\_Card CC

WHERE NOT EXISTS(SELECT Cust.\*

FROM Sale S

INNER JOIN [Check] C

ON C.check\_number=S.check\_number

INNER JOIN Customer\_Card Cust

ON Cust.card\_number=C.card\_number

WHERE Cust.cust\_surname=**CustSurname**

AND UPC NOT IN(SELECT UPC

FROM Sale SS

INNER JOIN [Check] CCC

ON CCC.check\_number=SS.check\_number

INNER JOIN Customer\_Card CustС

ON CustС.card\_number=CCC.card\_number

WHERE CustС.card\_number=CC.card\_number

)

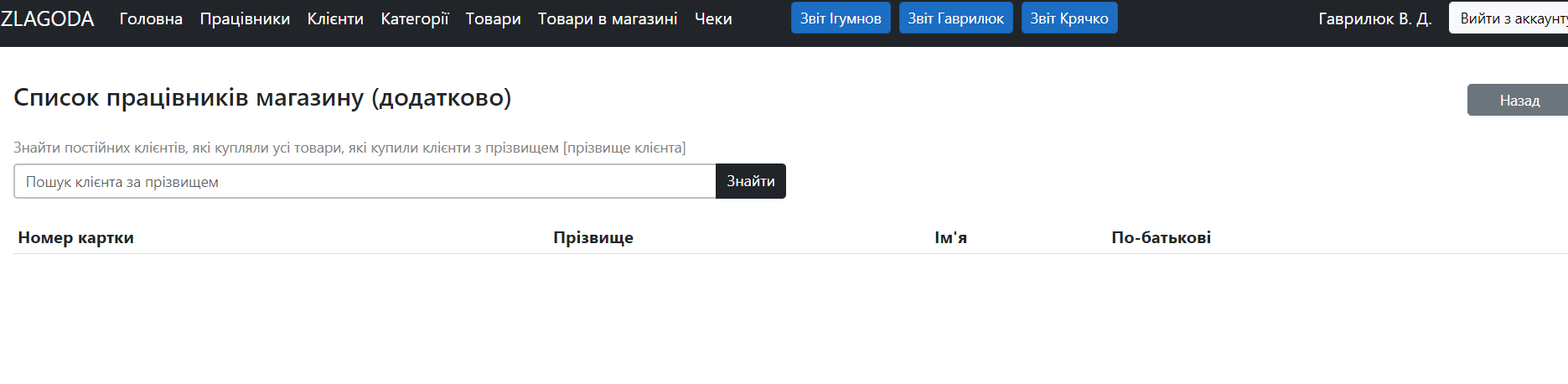
)

AND EXISTS (SELECT \*

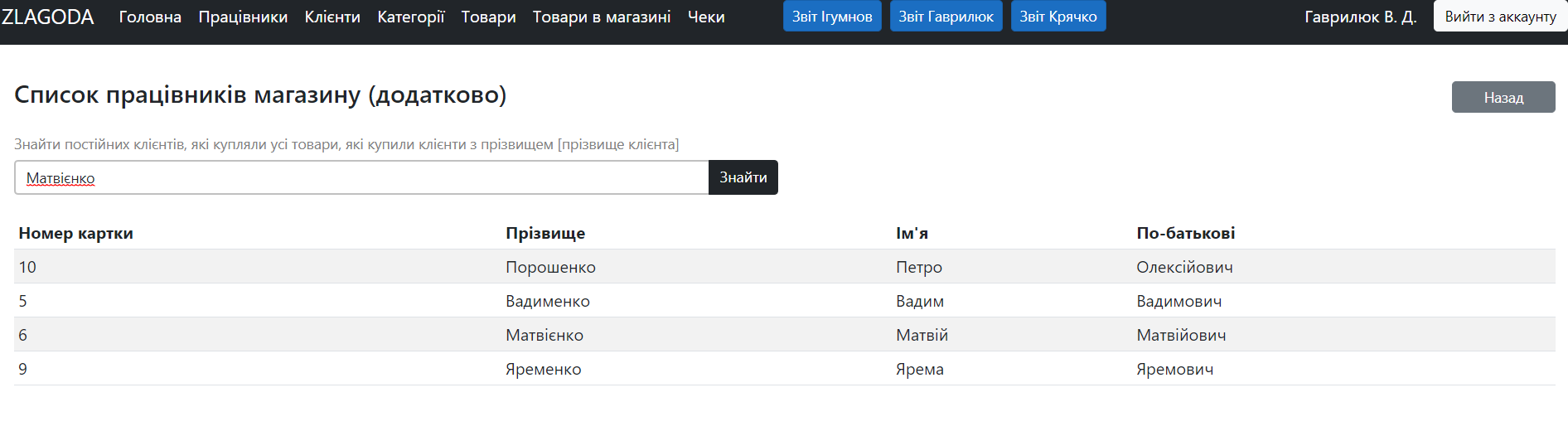
FROM Customer

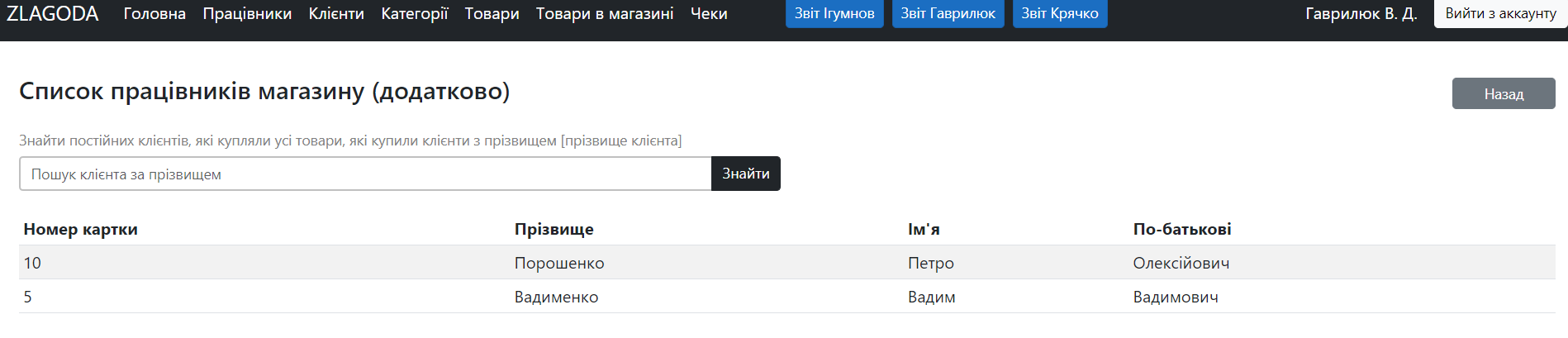
WHERE cust\_surname=**CustSurname**

)

*Скріншот з інтерфейсу користувача(в полі вводимо прізвище користувача):*

*Результат запиту(список всіх покупців,що підходять умові):*

Перший результат

Другий результат

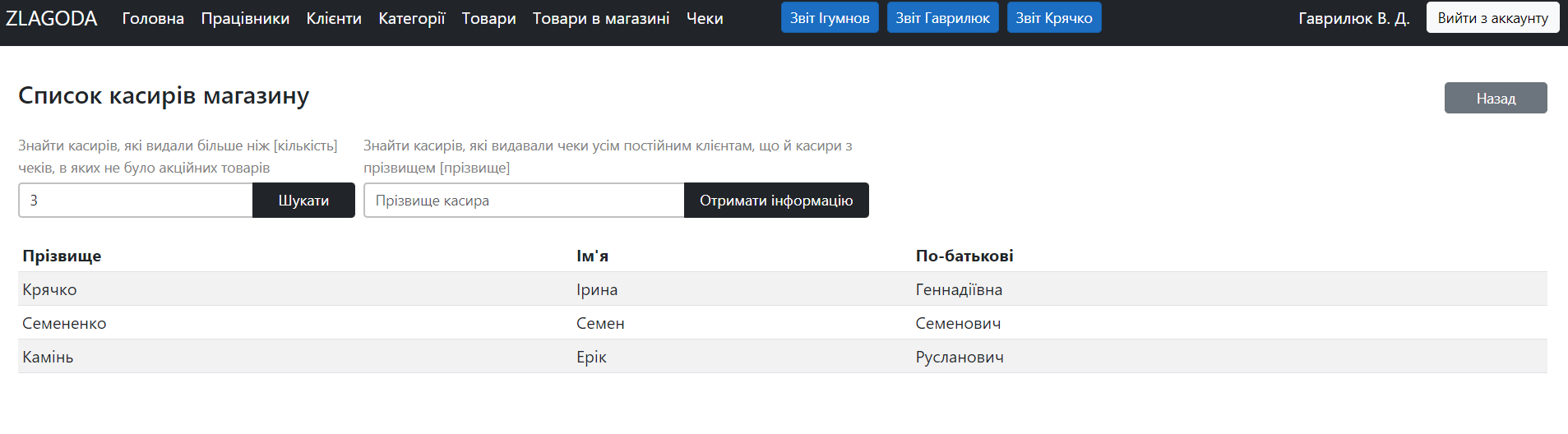
Як результат запит,ми отримуємо ім’я та номер картки покупців,що купували всі товари покупця,прізвище якого ми ввели в поле.

Код прикладної програми

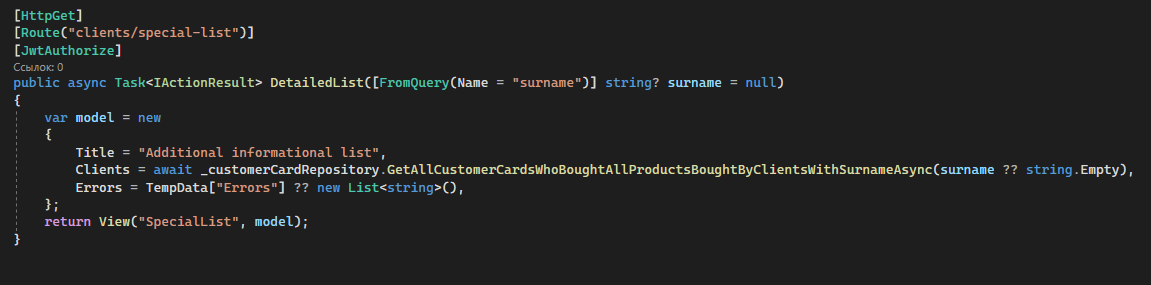
Розглянемо взаємодію коду з БД на прикладі першого запиту з групуванням.

1. При введенні числа в поле,до URL-адреси додається кількість чеків ,що ми ввели ,звертаючись до контролеру *DetailedList* ,що повертає результат виконаного запиту.





1. На рівні програмного коду,спочатку ми звертаємось до контролера *DetailedList,*котрий повертає HTML-сторінку,котра містить об’єкт з дод.інформацією та результат нашого запиту,який реалізований в асинхронному запиті *GetAllCustomerCardsWhoBoughtAllProductsBoughtByClientsWithSurnameAsync,*що приймає як параметр прізвище,котре ми ввели у формі ,показаній зверху.Контролер містить атрибути “Route” для вказання шляху ,за яким знаходиться сторінка з результатом запиту і JwtAuthorize,який відповідає за надавання певних прав окремим юзерам,наприклад менеджером.



1. **Для повернення результату контролером необхідно виконати метод *GetAllCustomerCardsWhoBoughtAllProductsBoughtByClientsWithSurnameAsync*,що реалізує запит до БД.Як параметр,він приймає прізвище,в змінній query знаходиться стрічка з SQL-кодом,який варто виконати.Далі через об’єкт connection здійснюємо під’єднання до БД,і через метод цього об’єкту *QueryAsync* виконуємо сам запит і повертаємо його результат у вигляді списку покупців.

Як результат,користувач отримує сторінку,на якій містяться всі покупці ,які купили ж ті продукти ,що і покупець з заданим прізвищем.