**Міністерство освіти і науки України**

**Чернівецький національний університет**

**імені Юрія Федьковича**

***Звіт***

про виконання лабораторної роботи № 2

з курсу «Інтелектуальні системи прийняття рішень»

Тема:

**Розробка статичної експертної системи направлено-мережевої структури**

Виконав:

студент 541 групи

Попелюк Михайло

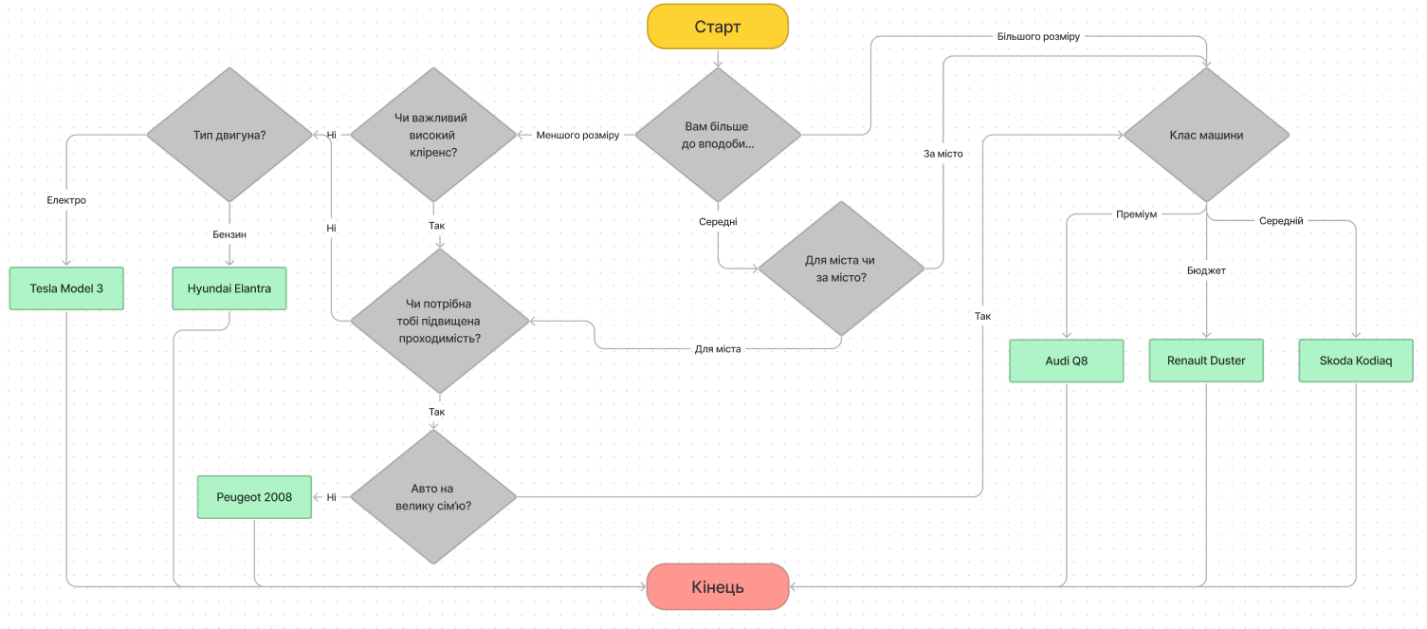
Чернівці 2022

**Завдання.** Побудувати дерево, яке перетинається направлено-мережевої структури. Побудова перехресних запитів.

Кінцевим результатом повинні бути звіт з описам етапів роботи та функціонуюча програма.

**Хід роботи**

У ході даної роботи було вирішено створити ЕС, яка допоможе користувачу обрати автомобіль, базуючись на його вподобаннях. Дерево перетинається, оскільки можна попасти на одну гілку кількома способами.

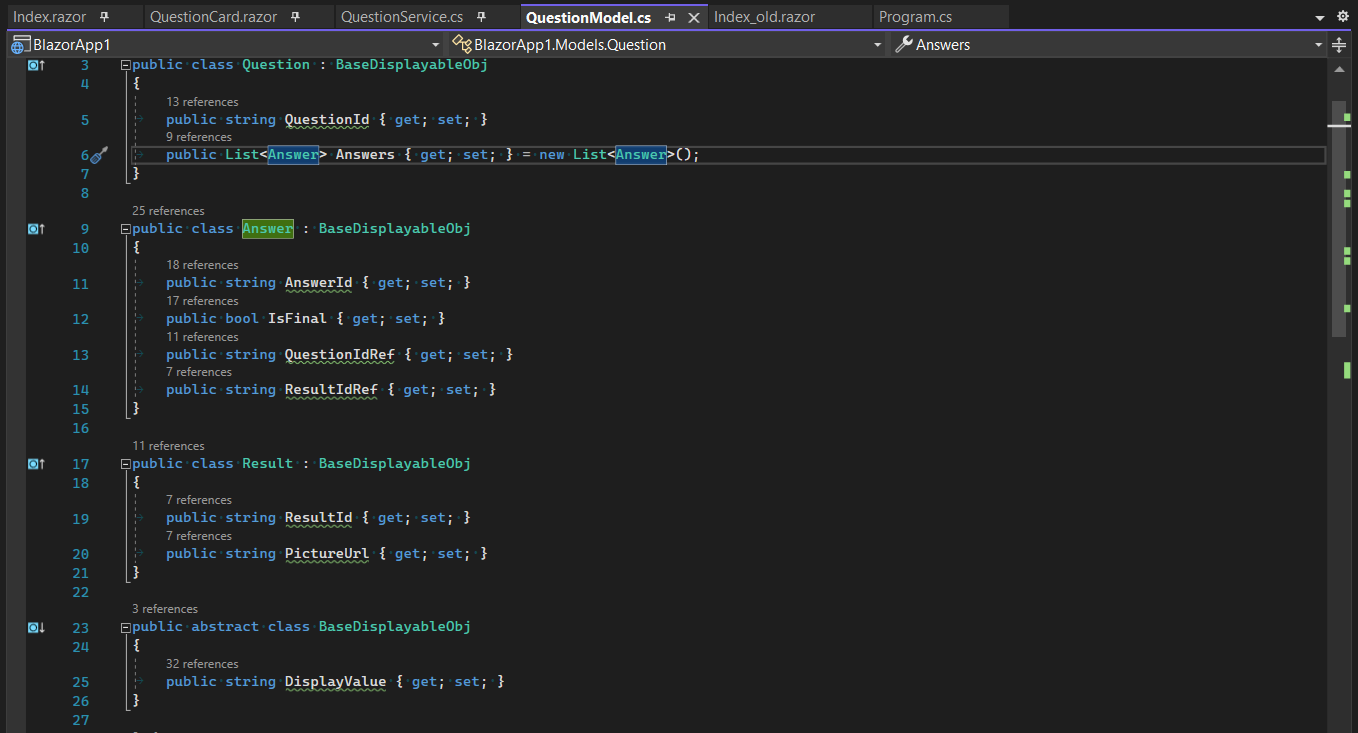


*Схема 1. Схема роботи експертної системи. Створено в figma*[1]

Для реалізації було вирішено використовувати бібліотеку Blazor.

Встановлення на початкові налаштування описані у лабораторній № 1, тому, в даному звіті ці кроки пропустимо.

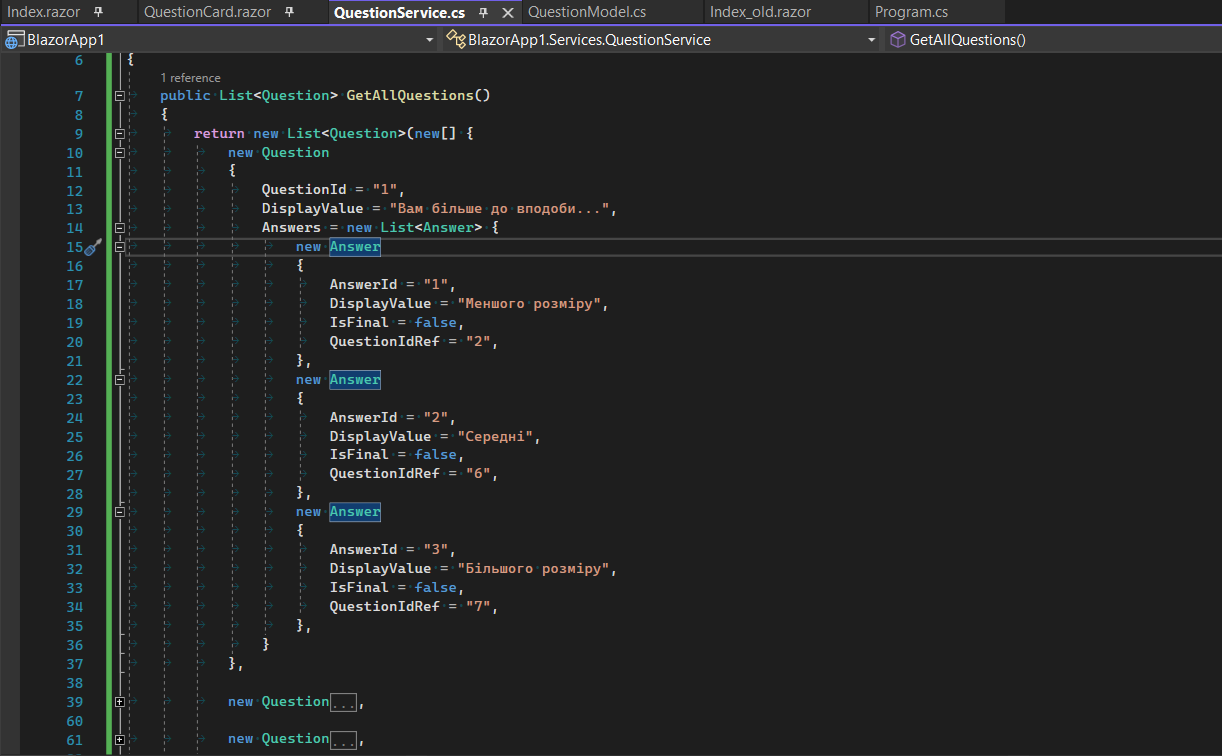
Для реалізації було обрано підхід, який передбачає створення об’єкту, який має унікальний ID питання, саме питання, список відповідей та посилання на наступне питання в залежності від відповіді.



*Рис. 1. Набір потрібних моделей класів*

Клас Question містить список відповідей (Answer) кожна з яких може бути як проміжним кроком так і фінальною відповідю, яка приведе до результату. Якщо відповідь буде проміжною то ID наступного питання буде міститись у полі QuestionIdRef. Якщо відповідь фінальна то ID результату буде міститись в ResultIdRef.

Створимо QuestionService у якому будуть міститись методи для отримання списку всіх питань і також списку з усіма результатами. Метод «GetAllQuestions» повертає всі можливі запитання з дерева, їх відповіді та відповідні ID в кожних відповідях. Метод «GetAllResults» повертає всі можливі результати, а також посилання на зображення моделі авто.



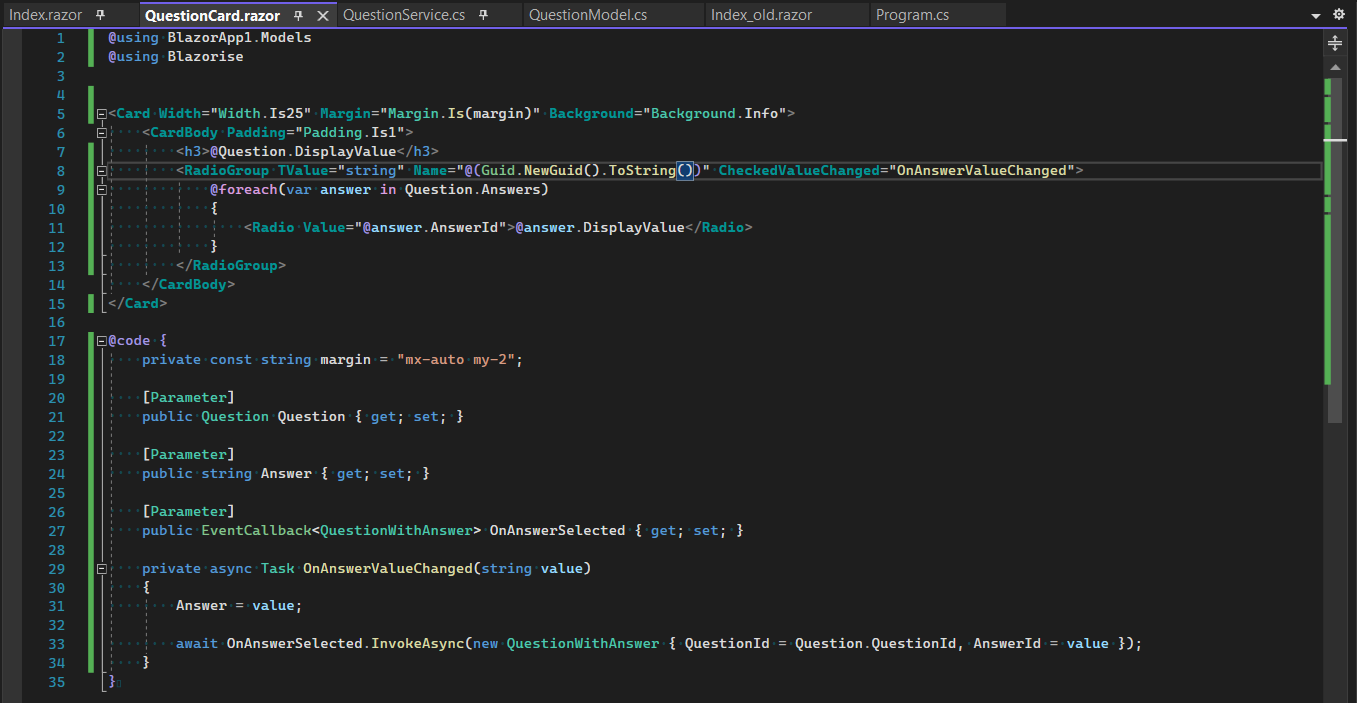
*Рис. 2. QuestionService і метод для отримання всіх питань*

*A screenshot of a computer

Description automatically generated*

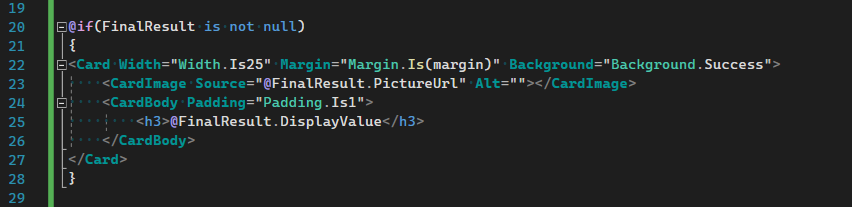
*Рис. 3. QuestionService і метод для отримання всіх результатів*

Тепер створимо QuestionCard.razor елемент, який вміє будувати візуальну частину одного питання (Question).



*Рис. 4. Візуальна частина для рендеру одного питання*

Добавимо реалізацію для відображення одного фінального результату.



*Рис. 5. Візуальна частина для рендеру фінального результату*

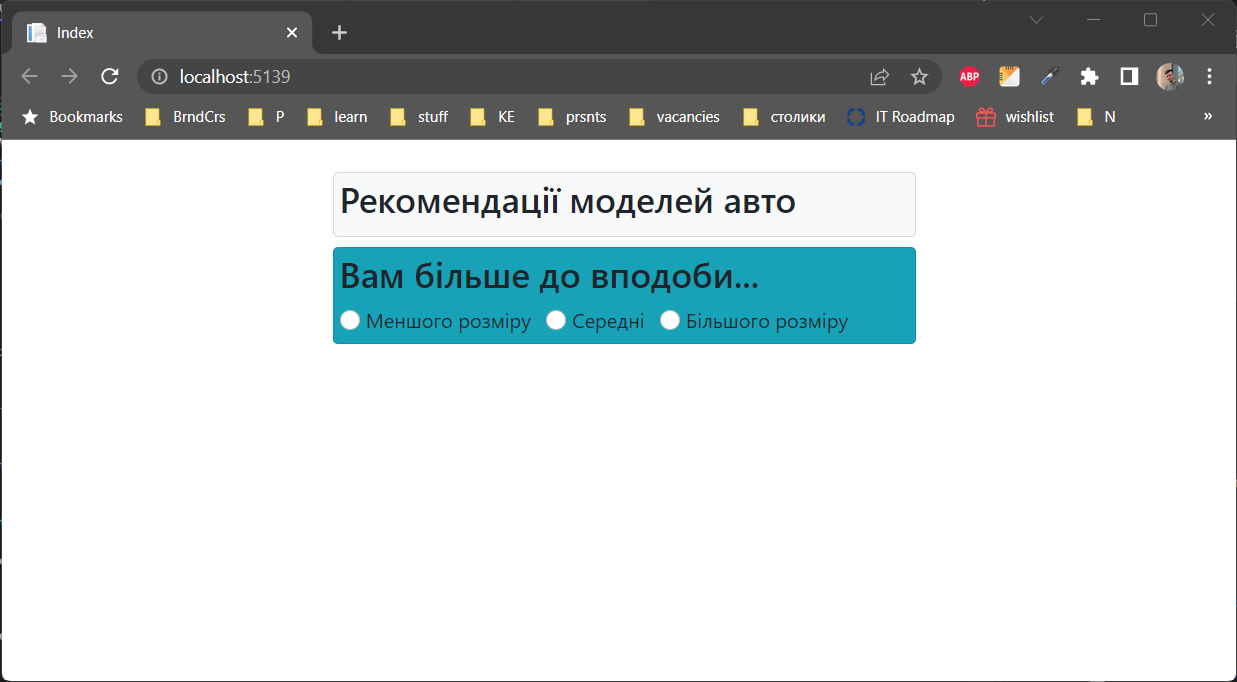
Тепер добавимо функціональну логіку яка обробляє результат вибору відповіді та показує наступне питанння чи фінальну відповідь.

Text

Description automatically generated

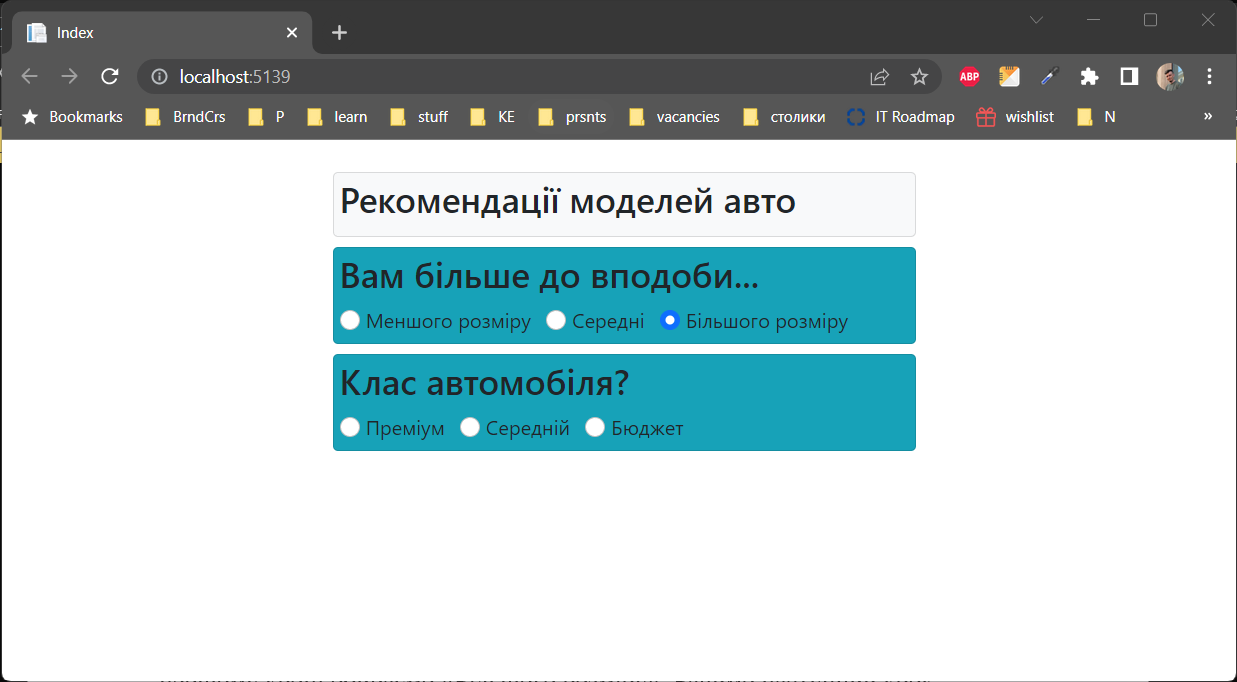
*Рис. 6. Візуальна частина для рендеру фінального результату*

Спробуємо запустити застосунок. Бачимо початкове вікно з першим питанням.



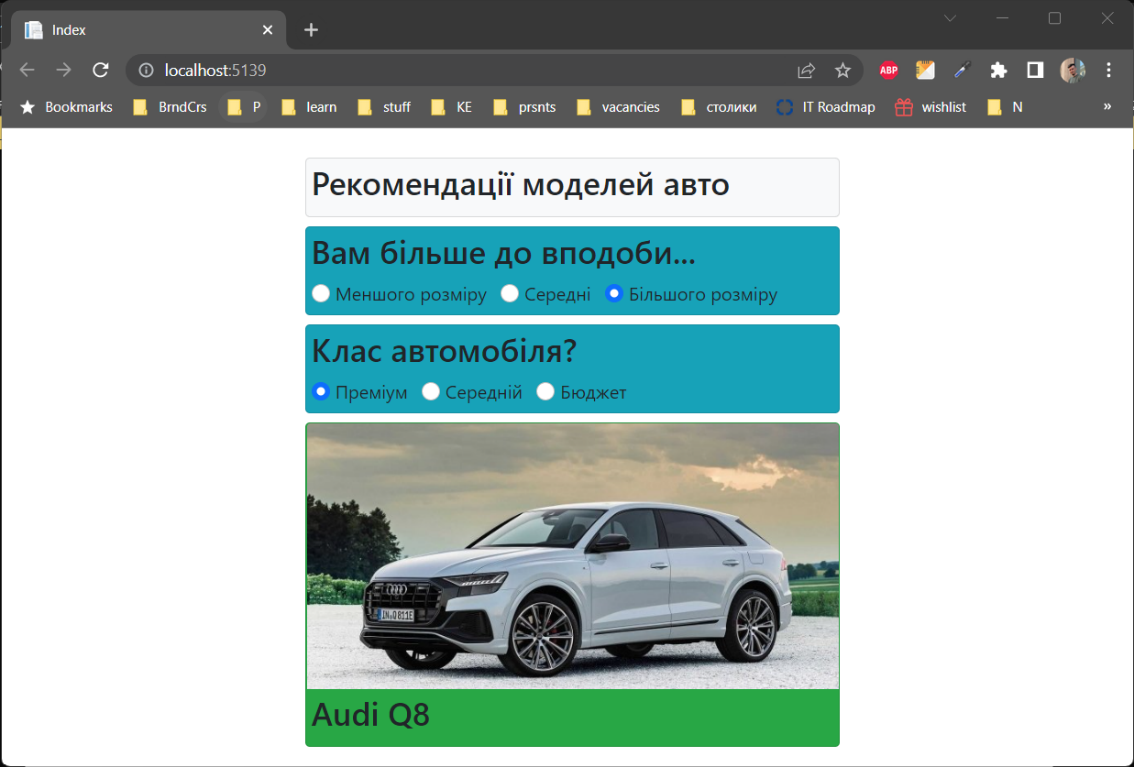
*Рис. 7. Початкове вікно*

Пройдемось по дереву так, щоб в кінці отримати «Audi Q8». Для цього на першому кроці обираємо «Більшого розміру». Бачимо наступний крок:



*Рис. 6. Другий крок*

Обираємо «Преміум» та отримуємо очікуваний результат «Audi Q8».



*Рис. 7. Відображення очікуваного результату*

Бачимо результат, який ми і очікували. Отже дерево реалізовано правильно.

**Тестування**

Розпочнемо тестування всіх можливих варіантів проходження по дереву рішень.

**Graphical user interface, text

Description automatically generated**

*Рис. 8. Тестування варіант 1*

*Graphical user interface, text

Description automatically generated*

*Рис. 9. Тестування варіант 2*

*Graphical user interface, text, website

Description automatically generated*

*Рис. 10. Тестування варіант 3*

Спробуємо отримати результат з тестування варіанту 3 перехресним запитом.

Graphical user interface, text

Description automatically generated

*Рис. 11. Тестування перехресний запит*

Спробуємо ще декілька різних варіантів проходу по дереві.

*Graphical user interface, website

Description automatically generated*

*Рис. 12. Тестування варіант 5*

*Graphical user interface, website

Description automatically generated*

*Рис. 13. Тестування варіант 6*

*Graphical user interface, website

Description automatically generated*

*Рис. 14. Тестування перехресний запит варіант 2*

Отже, після тестування застосунку, бачимо очікуваний результат. Тому можна дійти висновку, що наша експертна система працює відповідно до створеної схеми.

Весь код можна переглянути у GitHub репозиторії за посиланням у джерелах.[2]

**Висновок.** У ході даної лабораторної роботи мені вдалось побудувати дерево, яке перетинається та має направлено-мережеву структуру та видає результат на основі умов, що прописані у схемі її роботи. Користуючись деревом можна виконувати перехресні запити.

Джерела

1. Figma: <https://www.figma.com/file/85e1TTwrAJJEgCfbNA99ke/Untitled?node-id=0%3A1&t=oVyUl6NfGPsJ6Bo0-1>
2. GitHub: <https://github.com/mpopeljuk/uni-decision_making-lab_2>