НОВІ МОДЕЛІ ДЛЯ ПОШУКУ ОПТИМАЛЬНИХ ЦИКЛІВ В ГРАФІ ПРИ СПЕЦІАЛЬНИХ ОБМЕЖЕННЯХ: ПРИКЛАД ЗАСТОСУВАННЯ ДЛЯ ПІШОГО ТУРИЗМУ

Ідеатон «TRUE наука: знайомство та практика досліджень»
9 лютого 2024 року, м. Київ

Виконали: **Корабльов М.М.**[1,2] m.m.korablov@gmail.com,

Стоян О.О.[3] casperr3848@gmail.com,

Губернатор О.А.[3] o.a.huber@gmail.com

Науковий керівник: **Стецюк П.І.**[1,2] stetsyukp@gmail.com

---

[1]*Київський академічний університет*

[2]*Інститут кібернетики імені В.М. Глушкова НАН України*

[3]*Київський національний університет імені Тараса Шевченка*

Ідеатон «TRUE наука: знайомство та практика досліджень»
9 лютого 2024 року, м. Київ

Вступ

- Актуальність:

  Відома задача комівояжера має багато прикладних застосувань у логістиці та часто виникає як підзадача у багатьох інших сферах.
  Для розв’язання багатьох інших практичних завдань цієї моделі не достатньо і виникає потреба в розробці більш загальних математичних моделей.

- Мета:

  Ознайомитися з відомими узагальненнями задачі комівояжера та знайти напрямок, який потребує вдосконалення.

- Завдання:

  Розробити та протестувати власну модель, яка дозволить розв’язувати певний клас спеціальних задач планування оптимальних маршрутів.

Зміст

- Задача комівояжера та її узагальнення:
  - Постановка задачі;
  - Задача пошуку k-вершинного циклу, задача Set TSP.
- Найкоротший цикл через задану кількість вершин в кластерах графа:
  - Постановка задачі;
  - Математичні моделі на основі двох типів обмежень;
  - Приклад розв’язку задачі.
- Прототип додатку для піших прогулянок центром Києва:
  - Frontend складова, приклад роботи;
  - Backend складова, огляд архітектури;
  - Подальші удосконалення функціоналу.

ЗАДАЧА КОМІВОЯЖЕРА ТА ЇЇ УЗАГАЛЬНЕННЯ

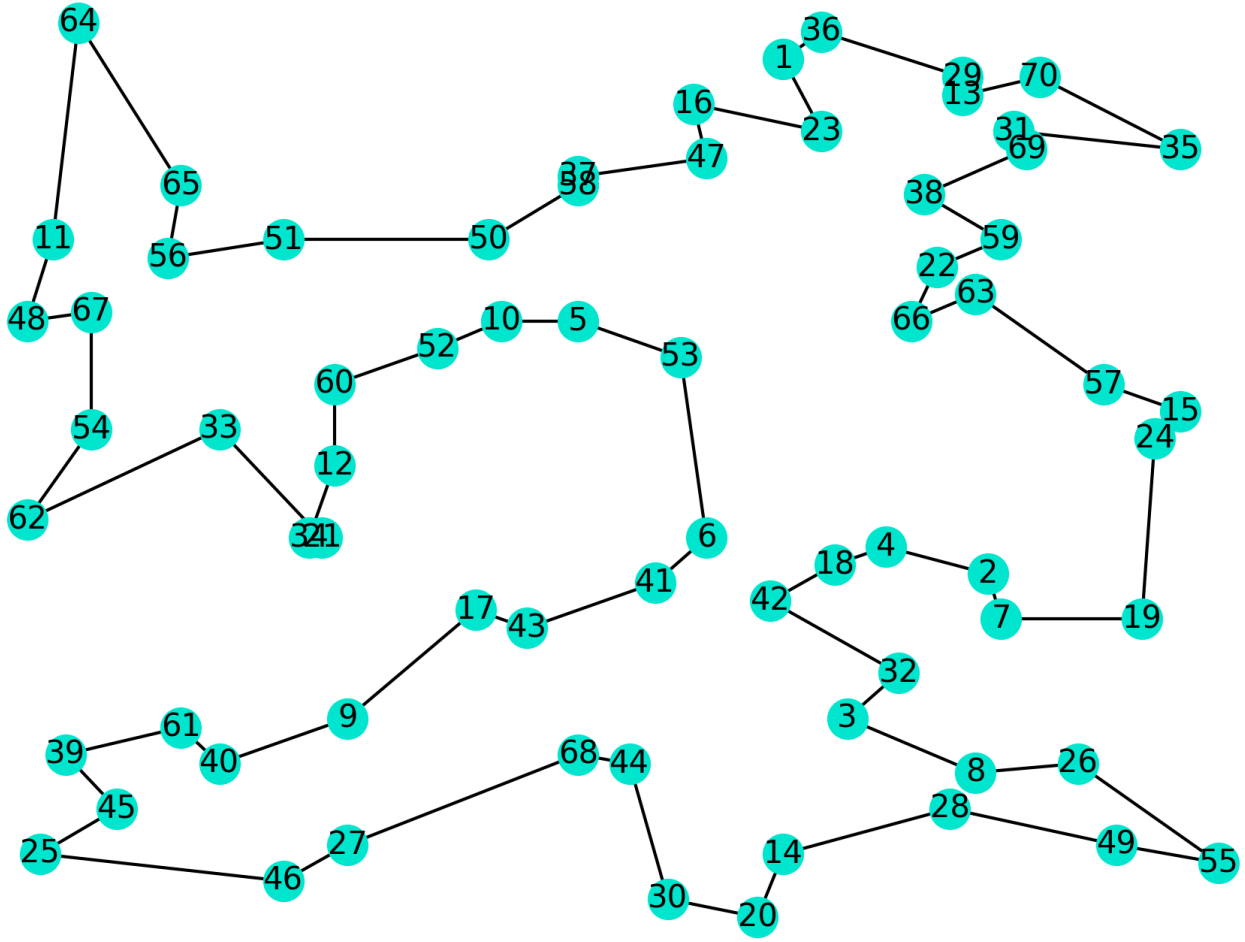
Задача комівояжера

- Нехай $D_{n,n}$ – повний граф, де n – кількість вершин, а $d_{ij} > 0$ – довжина дуги між вершинами i та j, де $i \neq j$.
- Зафіксуємо вершину s, яка буде початковою вершиною в графі.
- Цикл, який починається у вершині s і проходить через всі інші $n - 1$ вершин рівно один раз називається гамільтоновим. Найкоротший такий цикл і буде розв'язком задачі комівояжера.
- Серед математичних постановок даної задачі добре відомими та часто застосовними є моделі цілочислового лінійного програмування з обмеженнями Міллера, Такера і Земліна [1] та Гевіша і Грейвса [2].

[1] Miller C.E., Tucker A.W., Zemlin R.A. Integer programming formulation of travelling salesman problem. – J. ACM, 1960, 3, P. 326-329.

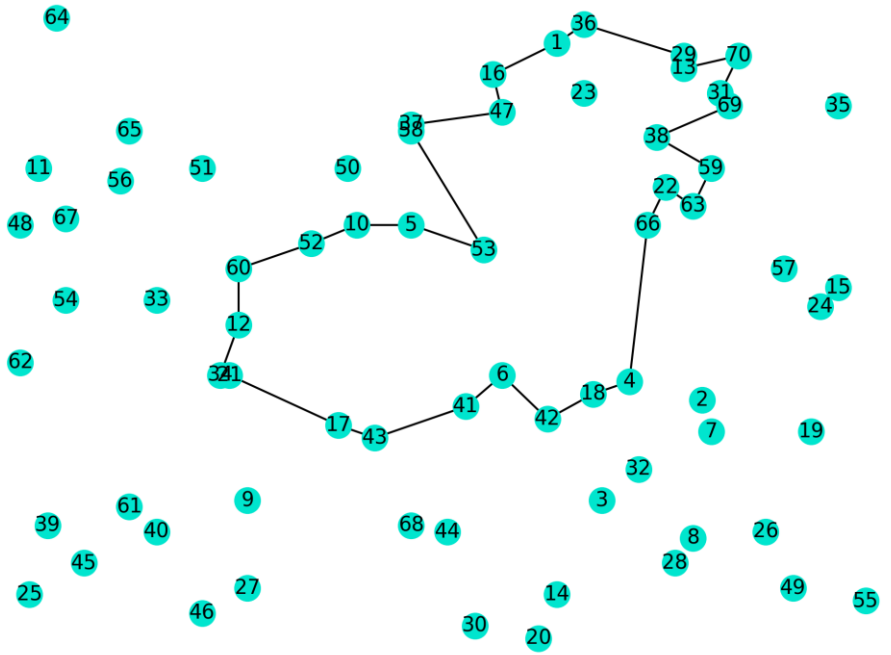
[2] Gavish B., Graves S.C. The travelling salesman problem and related problems. – Working Paper OR-078-78, 1978. – Operations Research Center, MIT, Cambridge, MA.

Приклад: граф st70.tsp, бібліотека TSPLib



Задача пошуку k-вершинного циклу

- В Інституті кібернетики були розроблені математичні моделі, які дозволяють шукати найкоротші цикли та шляхи, що проходять через $k < n$ вершин, накладаючи на них обмеження обов'язкового відвідування певних вершин [3].



[3] Стецюк П.І., Соломон Д.І., Григорак М.Ю. Задачі про найкоротші k-вершинні цикли та шляхи. Cybernetics and Computer Technologies. 2021. 3. С. 15–33.

Задача Set TSP

- Також існує певне узагальнення класичної задачі комівояжера, відоме під назвою Set Travelling Salesman Problem (Set TSP).

  Всі вершини графа, окрім стартової, розбиваються на K множин (кластерів), що не перетинаються. Задача полягає в пошуку найкоротшого циклу, який відвідає рівно по одній вершині з кожного кластеру, та повернеться до стартової вершини [4].

- ***N.B.*** Дана задача фактично є підвидом задачі пошуку K-вершинного циклу зі спеціальними обмеженнями на вибір вершин.

[4] Noon C.E. The generalized traveling salesman problem. Ph. D. Dissertation, University of Michigan, 1988

НАЙКОРОТШИЙ ЦИКЛ ЧЕРЕЗ ЗАДАНУ КІЛЬКІСТЬ ВЕРШИН В КЛАСТЕРАХ ГРАФА

Постановка задачі

- Нехай $D_{n,n}$ – повний граф, де n – кількість вершин, а $d_{ij} > 0$ – довжина дуги між вершинами i та j, де $i \neq j$.
- Зафіксуємо вершину s в графі, яка буде початковою.
- Інші вершини розділимо на кластери $C_k, k = 1, \dots, K$, що не перетинаються та позначимо через m_k кількість різних вершин з кожного кластера, які потрібно відвідати, $0 \leq m_k \leq |C_k|$.
- Потрібно знайти найкоротший цикл, який починається і закінчується в вершині s та проходить рівно через $M = \sum_{k=1}^{K} m_k$ вершин, по m_k з кожного кластера C_k.

- ***N.B.*** При $K = 1$ та $m_1 = n - 1$ отримуємо звичайну задачу комівояжера.

Модель на основі обмежень Міллера, Такера і Земліна

$$d_M^* = \min_{x_{ij}, y_i, u_i} \left\{ \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} d_{ij} x_{ij} \right\}$$

$$\sum_{j=1, j \neq s}^{n} x_{sj} = 1, \quad \sum_{j=1, j \neq i}^{n} x_{ij} = y_i; \qquad i \in \{1, \dots, n\}, i \neq s$$

$$\sum_{j=1, j \neq s}^{n} x_{js} = 1, \quad \sum_{j=1, j \neq i}^{n} x_{ji} = y_i; \qquad i \in \{1, \dots, n\}, i \neq s$$

$$\sum_{i \in C_k} y_i = m_k; \qquad k \in \{1, \dots, K\}$$

$$u_i - u_j + M x_{ij} \leq M - 1; \qquad i, j \in \{1, \dots, n\}, i \neq j, i \neq s, j \neq s$$

$$x_{ij} = 0 \vee 1; \qquad i, j \in \{1, \dots, n\}, i \neq j$$

$$y_i = 0 \vee 1; \qquad i \in \{1, \dots, n\}, i \neq s$$

$$1 \leq u_i \leq M; \qquad i \in \{1, \dots, n\}, i \neq s$$

Модель на основі обмежень Гевіша і Грейвса

$$d_M^* = \min_{x_{ij}, y_i, z_{ij}} \left\{ \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} d_{ij} x_{ij} \right\}$$

$$\sum_{j=1, j \neq s}^{n} x_{sj} = 1, \quad \sum_{j=1, j \neq i}^{n} x_{ij} = y_i; \qquad i \in \{1, \dots, n\}, i \neq s$$

$$\sum_{j=1, j \neq s}^{n} x_{js} = 1, \quad \sum_{j=1, j \neq i}^{n} x_{ji} = y_i; \qquad i \in \{1, \dots, n\}, i \neq s$$

$$\sum_{i \in C_k} y_i = m_k; \qquad k \in \{1, \dots, K\}$$

$$z_{ij} - M x_{ij} \leq 0; \qquad i, j \in \{1, \dots, n\}, i \neq j$$

$$\sum_{j=1, j \neq s}^{n} z_{sj} = M, \quad \sum_{j=1, j \neq s}^{n} z_{js} = 0$$

$$\sum_{j=1, j \neq i}^{n} z_{ij} - \sum_{j=1, j \neq i}^{n} z_{ji} = -y_i; \qquad i \in \{1, \dots, n\}, i \neq s$$

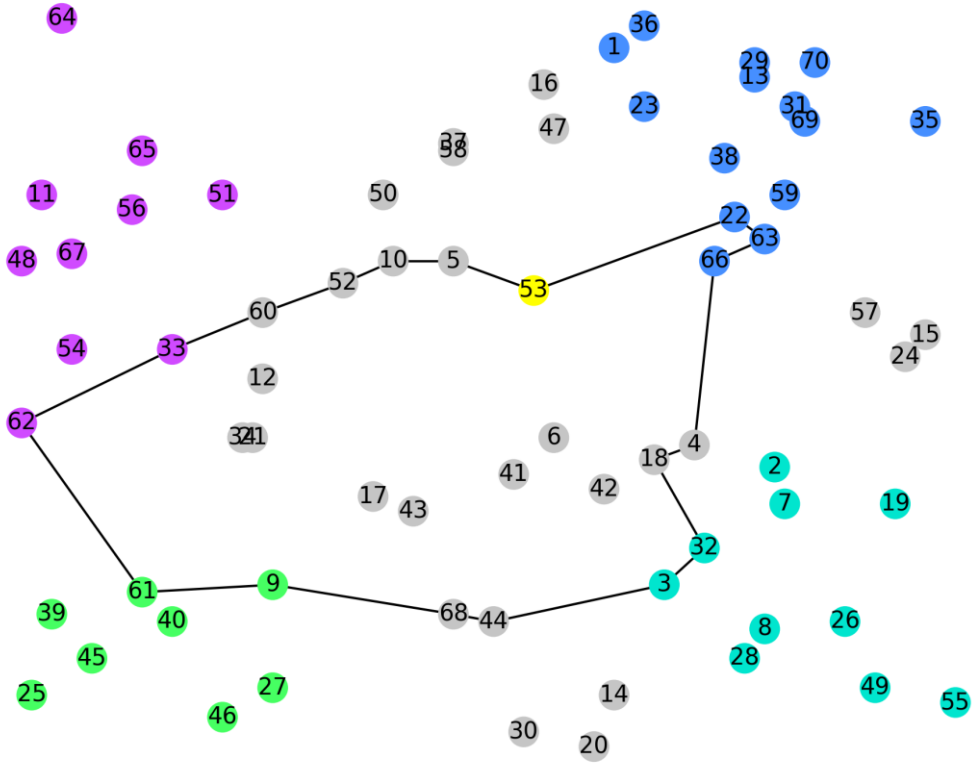
$$x_{ij} = 0 \vee 1; \qquad i, j \in \{1, \dots, n\}, i \neq j$$

$$y_i = 0 \vee 1; \qquad i \in \{1, \dots, n\}, i \neq s$$

$$z_{ij} \geq 0; \qquad i, j \in \{1, \dots, n\}, i \neq j$$

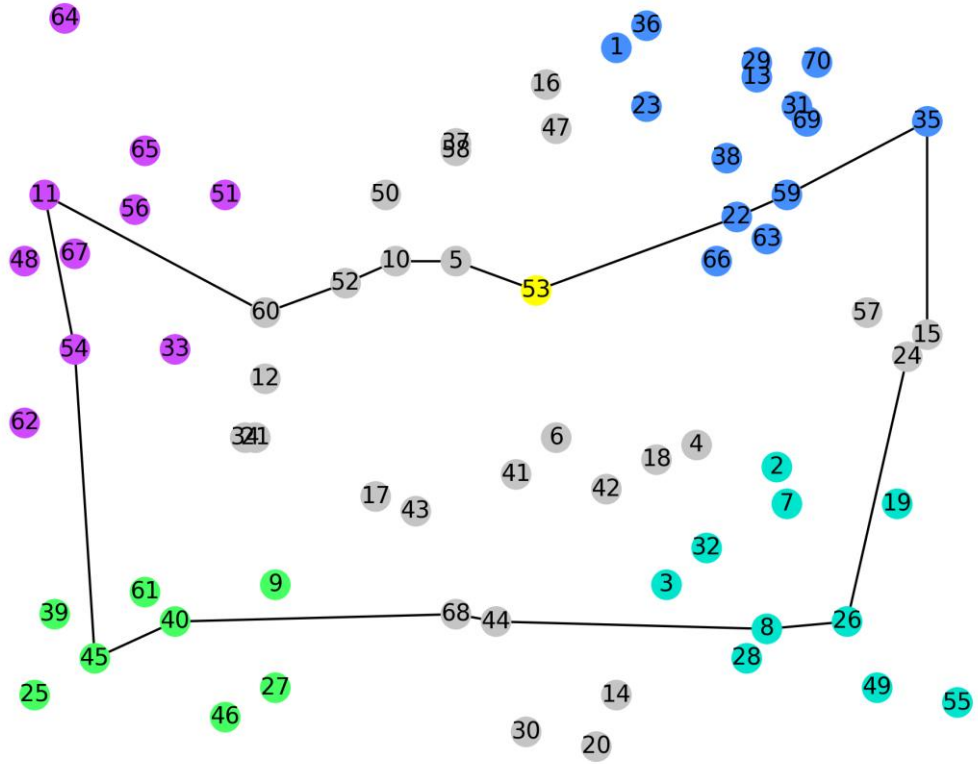
Приклад: граф st70.tsp

- Початок у s=53, потрібно відвідати: 3 сині, 2 фіолетові, 2 зелені, 2 бірюзові, 8 сірих



$\boldsymbol{d^{*}_{17} = 217}$

можна відвідувати довільні вершини



$\boldsymbol{d^{**}_{17} = 310}$

обов’язкове відвідування вершин 11, 26, 35, 45

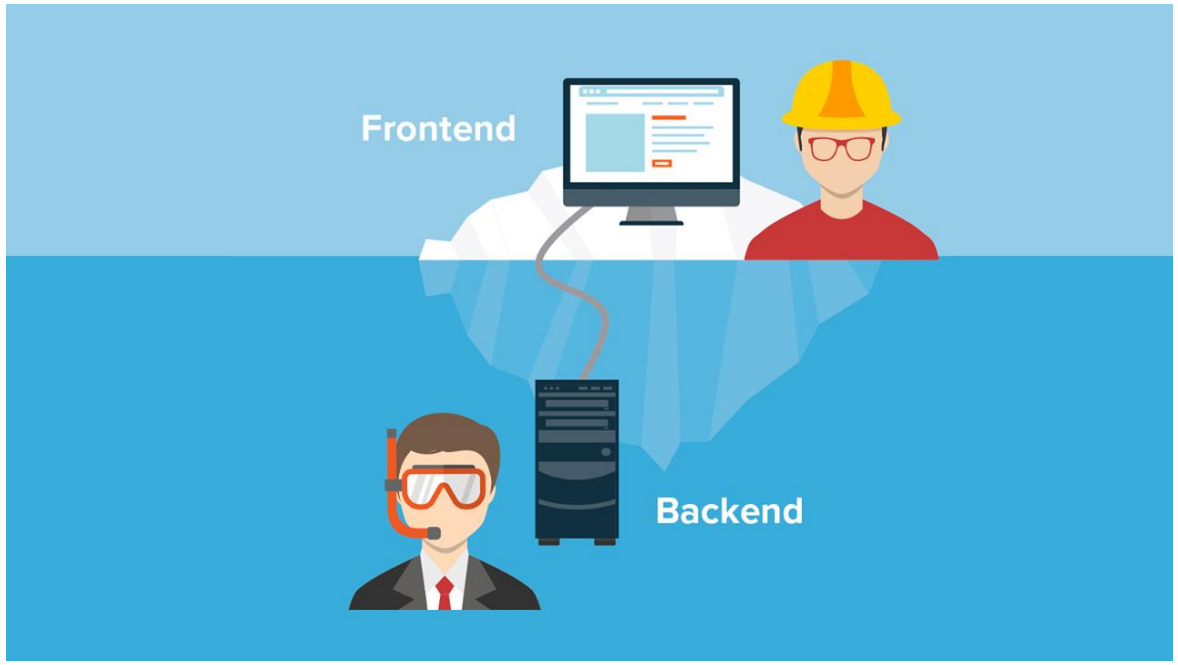
ПРОТОТИП ДОДАТКУ ДЛЯ ПІШИХ ПРОГУЛЯНОК ЦЕНТРОМ КИЄВА

Застосування для туризму

- Створений веб-додаток вносить інновації в планування пішохідних маршрутів Києвом.
- Застосунок дозволяє з легкістю прокладати найкоротший маршрут, який відвідує задану кількість локацій різного виду.
- Мета проекту – забезпечити доступний, зручний і змістовний пішохідний туризм для всіх, хто відвідує столицю або проживає у ній.

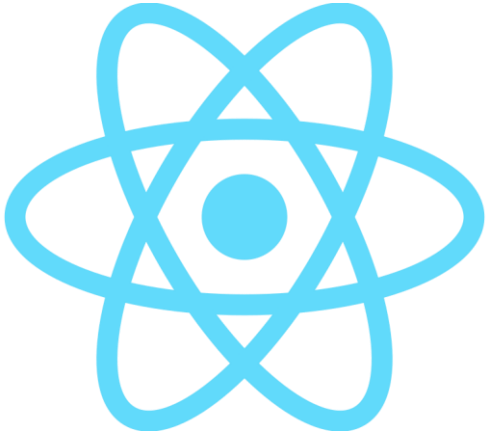
FrontEnd складова веб-додатку

- Frontend – це частина веб-додатку, з якою користувач безпосередньо взаємодіє. Він відповідає за відображення інтерфейсу, перетворюючи бізнес-логіку та дані з бекенду у зрозумілу та зручну форму.

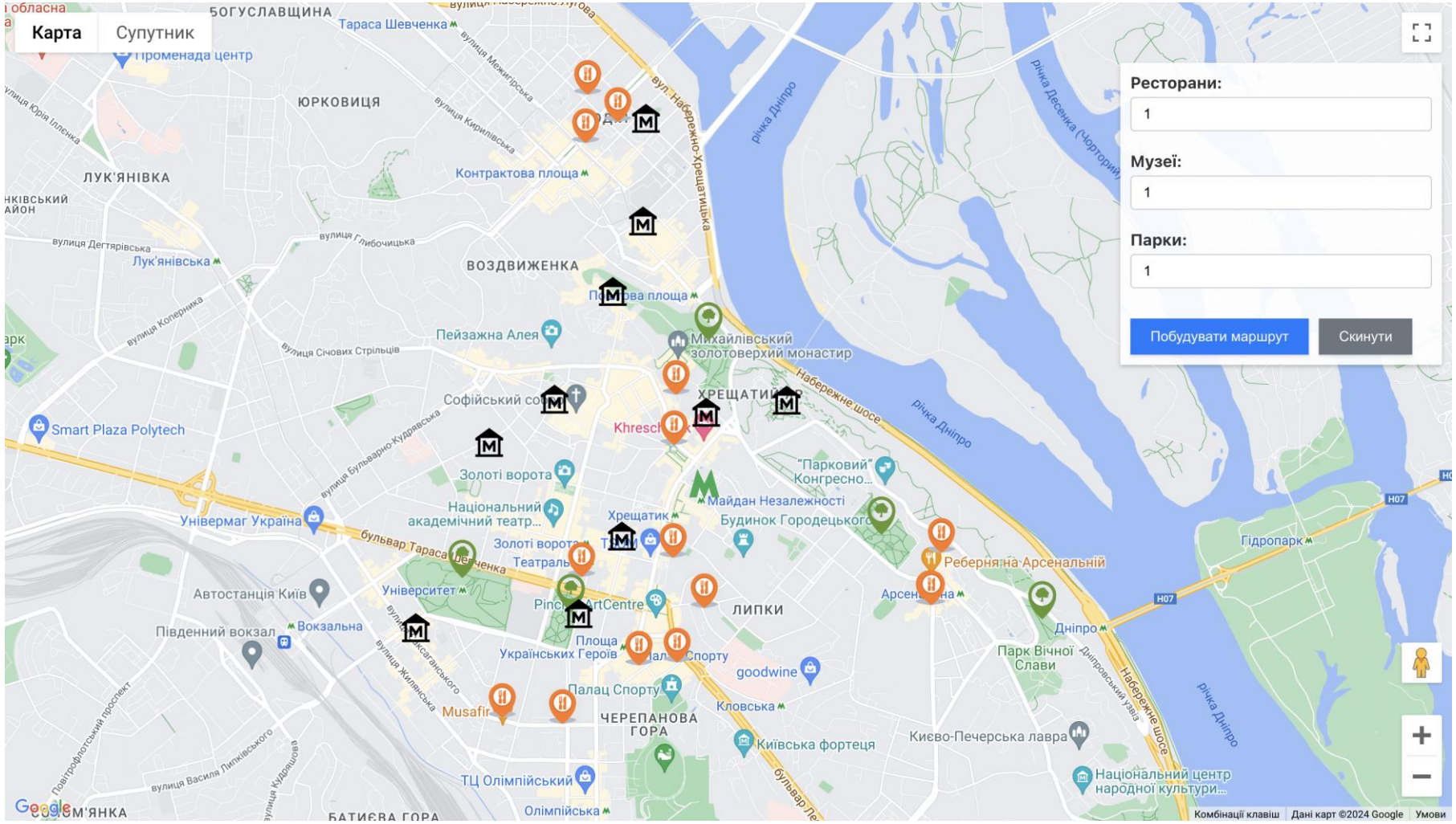


Основні технології

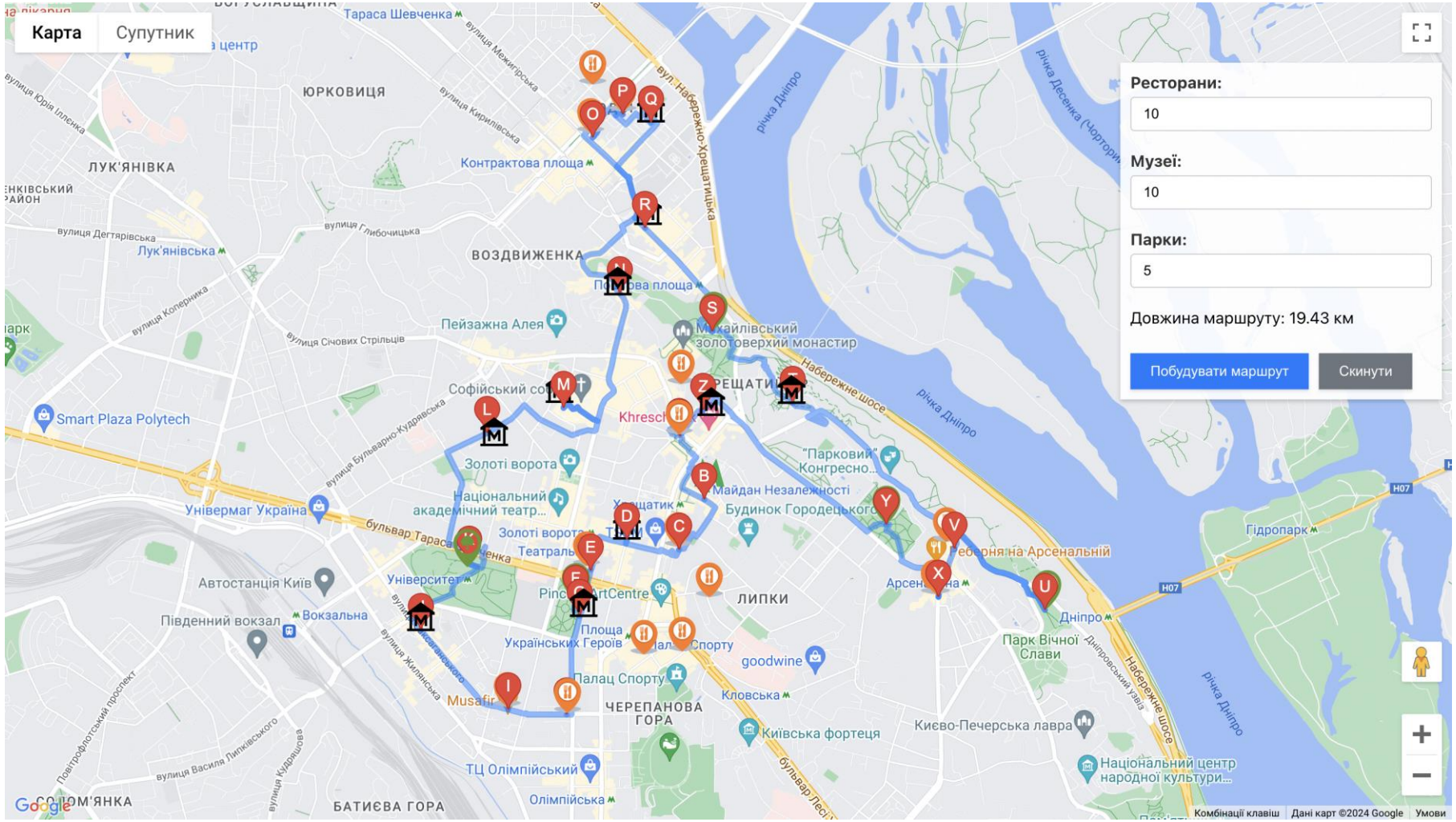
- У розробці веб-застосунку для пішохідного туризму використовуються сучасні технології, що дозволяють ефективно реалізувати функціонал маршрутизації та інтерактивної роботи з картою. Ключовими технологіями є:
  - **React:** Дозволяє створювати динамічний і зручний інтерфейс користувача;
  - **Google Maps API:** Надає можливості для інтеграції карт, міток та маршрутизації;
  - **Axios:** Використовується для здійснення HTTP-запитів до сервера для отримання та відправлення даних.



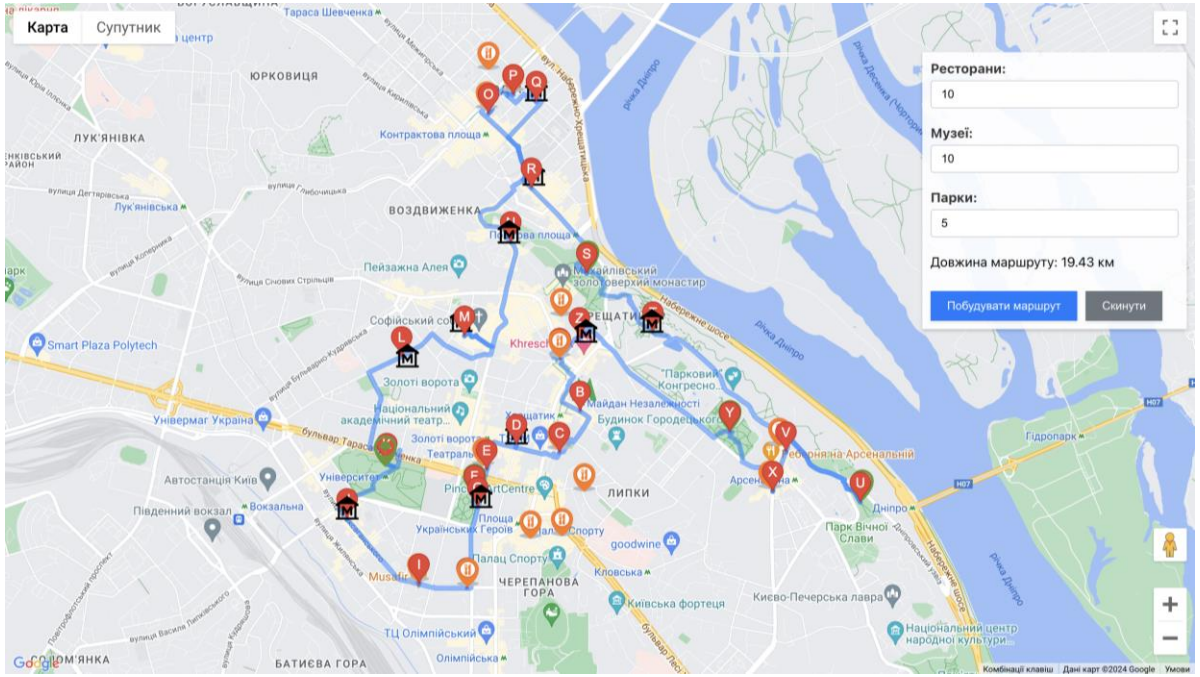
Інтерфейс користувача

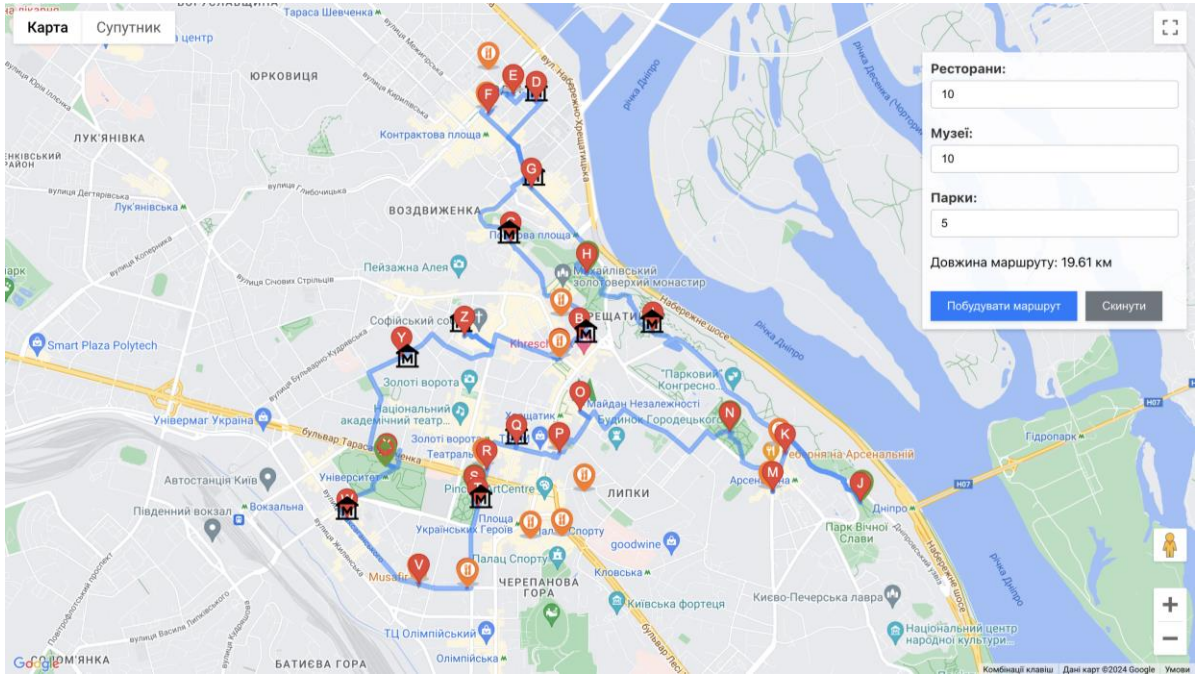


Приклад побудови маршруту (відстань «пішки»)



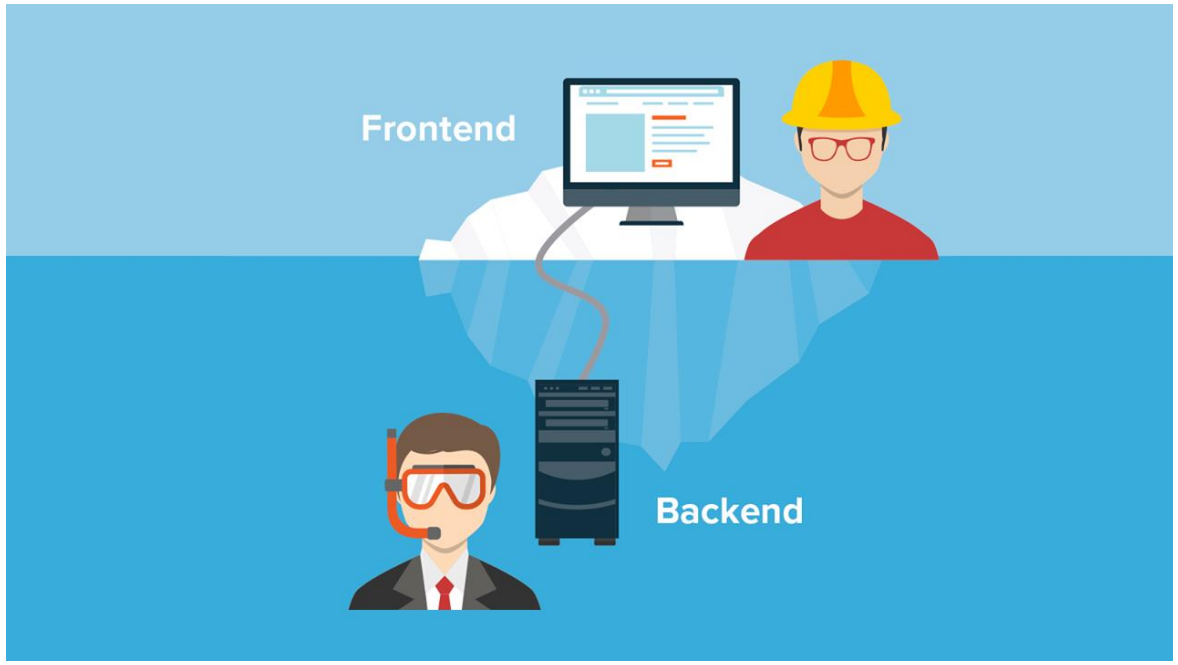
Оптимізація маршруту: наше рішення проти Google Maps





Backend складова веб-додатку

- Backend це розробка бізнес-логіки продукту (сайту або веб-додатку). Він відповідає за взаємодію користувача з внутрішніми даними, які відображає frontend.



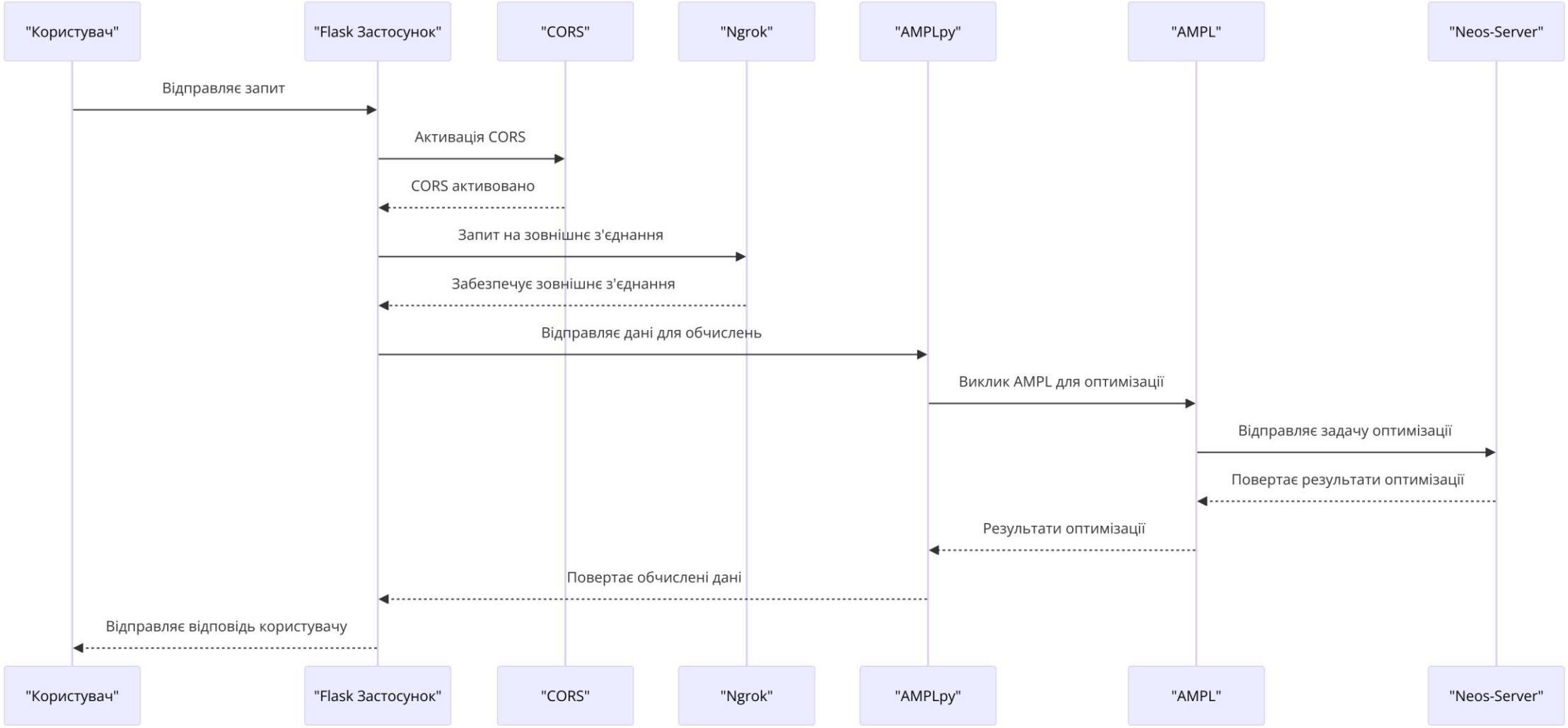
Основні технології розробки backend сервера

- **Flask:** Використовується для створення інтерфейсів API, обробки запитів та відповідей;
- **Flask-CORS:** Розширення для Flask, яке дозволяє обробляти Cross-Origin Resource Sharing (CORS) запити;
- **Ngrok:** Інструмент, що дозволяє тунелювати запити з Інтернету на локальний розробницький сервер.

Інструменти для розробки бізнес-логіки backend

- **Neos-server:** Для обчислення моделі використовується API (application programming interface) neos-server.org;
- **AMPL:** Мова алгебраїчного моделювання, на якій написана логіка знаходження оптимальних циклів в графі;
- **Amplpy:** Для обробки результатів та взаємодії з AMPL моделлю мовою програмування Python.

Архітектура обробки запитів в backend



Використання Google Colab для backend

```
# Встановлення необхідних бібліотек
!pip install flask_ngrok
!pip install pyngrok==4.1.1
!ngrok authtoken 2bPED8VS30QLXZT8XRtCipJmbXG_2oUFhebkYupMnbwTpLthN
!pip install flask-cors
#=================================ampl=================================
%pip install -q amplpy
# Google Colab & Kaggle integration
from amplpy import AMPL, ampl_notebook

ampl = ampl_notebook(
    modules=["coin", "highs", "gurobi", "gokestrel"],  # modules to install
    license_uuid="7a5e7dba-807d-4d3e-87bc-df541606f39b",  # license to use
)
```

```
from flask import Flask, request, jsonify
from flask_cors import CORS
from flask_ngrok import run_with_ngrok

app = Flask(__name__)
CORS(app)  # Включаємо CORS для всіх доменів на всіх маршрутах
run_with_ngrok(app)  # Запуск ngrok, коли Flask стартує
```

```
 * Running on http://127.0.0.1:5000
INFO:werkzeug:Press CTRL+C to quit
 * Running on http://bde8-34-90-148-134.ngrok-free.app
 * Traffic stats available on http://127.0.0.1:4040
```

Посилання

Подальші удосконалення функціоналу

- Можливість будувати не тільки цикл, а й найкоротший шлях з вершини A до вершини B, який має відвідати певну кількість вершин з різних кластерів;
- Фіксування конкретних вершин обов’язковими для відвідування;
- Задавання порядку відвідування деяких вершин;
- Знаходження різних найкоротших маршрутів, якщо такі є;
- Побудова маршруту для m комівояжерів при подібних обмеженнях;
- Створення користувачем своїх кластерів та додавання нових вершин до вже існуючих;
- Підтримка графів більшої розмірності завдяки застосуванню повної версії пакету AMPL, або ж іншого схожого обчислювального ядра.

Список джерел посилання

1. Miller C.E., Tucker A.W., Zemlin R.A. Integer programming formulation of travelling salesman problem. – J. ACM, 1960, 3, P. 326-329.
2. Gavish B., Graves S.C. The travelling salesman problem and related problems. – Working Paper OR-078-78, 1978. – Operations Research Center, MIT, Cambridge, MA.
3. Стецюк П.І., Соломон Д.І., Григорак М.Ю. Задачі про найкоротші k-вершинні цикли та шляхи. Cybernetics and Computer Technologies. 2021. 3. С. 15–33. https://doi.org/10.34229/2707-451X.21.3.2
4. Noon C.E. The generalized traveling salesman problem. Ph. D. Dissertation, University of Michigan, 1988
5. Fourer R., Gay D., Kernighan B. AMPL, A Modeling Language for Mathematical Programming. Belmont: Duxburry Press, 2003. 517 p.
6. NEOS Solver. https://neos-server.org/neos/solvers/

ДЯКУЄМО ЗА УВАГУ