

## Завдання півфіналу UA Web Challenge VIII **Front-end (JavaScript)** **Middle/Senior**

Усі учасники півфіналу працюють над частинкою одного проекту. Завдання Front-end (JavaScript) – надати зручний та якісний матеріал для подальшої Back-end розробки.

**Звертаємо увагу:** макет фіналіста обраний для цього завдання незалежно від рейтингу учасників номінації Front-end (HTML) та Web Design.

1. Завдання
2. Критерії оцінки
3. Формат рішення
4. Судді
5. Контакти

### 1. Завдання

Реалізувати FE (JavaScript) частину [онлайн-сервісу](#) у вигляді односторінкового веб-додатку для знаходження найбільш безпечного шляху для проїзду між двома точками на карті.

Замовник проекту:



**Департамент патрульної  
поліції МВС**

*«Нова поліція означає також і нові підходи до завдань аналітики. Легковажні, на перший погляд, ігри з числами перетворюються у серйозні рішення, що впливають на роботу поліцейських. Розстановка сил, підбір напарників, політики застосування сили – все це залежить від аналітичних висновків. Творчі підходи і системна праця завжди дасть результат. Будьте розумними й наполегливими. Слава Україні!»*

**Владислав Власюк,  
заступник начальника Департаменту патрульної поліції МВС**

**Деталі завдання:**

- 1.1. **Проблема, яка вирішується:** кожного року на певних ділянках доріг трапляються сотні ДТП. Як результат – жертви, затори, перевантаженість роботи поліції. Це можна попередити, якщо завчасно проінформувати водія про небезпеку, що може очікувати його протягом поїздки та допомогти обрати найбезпечніший маршрут.
- 1.2. **Специфікація поведінки юзера на сайті:**
  - (1) юзер заходить на сайт без реєстрації;
  - (2) вводить початкову та кінцеву точку маршруту;
  - (3) система виводить на екран альтернативні маршрути за рівнем безпеки у відповідному кольорі:
    - зелений – маршрут з найменшою кількістю небезпечних точок
    - жовтий – маршрут з середньою кількістю небезпечних точок
    - червоний – маршрут з найбільшою кількістю небезпечних точок.
  - (4) На кожному маршруті мають показуватися прапорці на точках ДТП та інформацією про кількість ДТП за рік, у тому числі з жертвами.
- 1.3. Приклад даних, які використовуватимуться [тут](#).
- 1.4. Для логіки роботи сайту з точки зору дизайнера, а також для отримання необхідних відсутніх елементів, можете використати макет, [підготовлений дизайнером](#).
- 1.5. Дані про кількість та місцезнаходження аварій будуть отримуватися від Back-end.
- 1.6. **Головна сторінка**
  - (1) На головній сторінці користувач повинен вказати адреси відправки і прибуття (або пропустити, перейшовши за посиланням "на мапі").
  - (2) При введенні адреси йде запит до сервера для отримання списку знайдених адрес, які потрібно показати користувачеві.
  - (3) Коли адреси обрані, кнопка "Показати маршрут" стане доступною і переведе користувача до екрану карти.
- 1.7. **Екран мапи**
  - (1) Додаток складається з полотна карти і бічного меню, перекриваючого його.
  - (2) Якщо адреси не вказані - меню заховано.
  - (3) Адресу можна додати кліком по карті або в списку адрес в бічному меню.
  - (4) Адреси можна видалити і поміняти місцями.
  - (5) Кожного разу, коли змінюються адреси, потрібно перемалювати маршрути на карті і місця ДТП.
  - (6) Сервер повертає не більше трьох маршрутів (безпечний, майже безпечний і небезпечний).
  - (7) Промальовування маршруту відбувається по точках, отриманих з сервера.
  - (8) Місця ДТП теж виходять з сервера і відзначаються іконками на мапі.
  - (9) Детальна інформація за маршрутами відображається в бічній панелі.
  - (10) Можна вимкнути маршрут, що б він не відображався на мапі.
  - (11) Детальна інформація про маршрут описує відстані і повороти.
  - (12) Користувач може в будь-який момент змінити маршрут: все має оновитися.
- 1.8. **Вимоги:**
  - (1) Хороша організація коду, модульна структура.
  - (2) Документування коду згідно синтаксису JSdoc.
  - (3) Використовувати систему збірки (gulp, grunt).
  - (4) Покрити тестами блоки роботи з сервером і користувачем.

- (5) Реалізувати модульний односторінковий додаток.
- (6) Зберігання стану і відновлення всіх полів при перезавантаженні сторінки.
- (7) Для зниження навантаження на сервер запити до нього необхідно ставити в чергу.
- (8) Завдання потрібно виконати на JavaScript, неприпустимо використання трансьльованих мов (CoffeeScript, LiveScript, Emscripten і т.д.).
- (9) Обмеження по бібліотеках і фреймворках відсутні.
- (10) Підтримка браузерів: основні + IE9.

## 2. Критерії оцінки:

- 2.1. Загальна архітектура додатку.
- 2.2. Чистота і ефективність коду.
- 2.3. Протокол обміну даними з сервером.
- 2.4. Робота з DOM елементами і API карти.
- 2.5. Повнота документації.
- 2.6. Ступінь покриття тестами.

## 3. Формат рішення

- 3.1. Рішення надсилати в **ОДНОМУ** файлі-архіві **виду та з назвою «UWCUA.zip»**. Будь ласка, оптимізуйте розмір архіву до **не більше 10 Мб**.
- 3.2. Організатори та судді залишають за собою право дискваліфікувати роботу учасника, якщо робота:
  - (1) містить будь-яку вказівку на ім'я, прізвище, електронну пошту, компанію, адресу чи інші персональні дані учасника;
  - (2) виконана у іншому форматі, ніж вказано у завданні;
  - (3) виконана з допомогою сторонніх осіб, а не учасником особисто;
  - (4) якщо робота буде виявлена у відкритому доступі до моменту закінчення прийому робіт.

## 4. Судді



**Олександр Тарасюк**

Розробник програмного забезпечення. Більше 6 років досвіду в розробці комерційного програмного забезпечення в якості Front-End і Back-end розробника. Спеціалізація: Front-End розробка, оптимізація продуктивності. Технології і мови - JavaScript, Node.js, HTML, CSS.



**Максим Климишин**

CTO в компанії GVMachines Inc. (Zakaz.ua, CartFresh.com) Інженер з 12-річним досвідом роботи над e-commerce та консюмерськими проектами. Основні мови – Python, JavaScript, Clojure. Напрямки роботи – архітектура високонавантажених систем, розподіленні обчислення та зберігання даних, підходи в розробці клієнтських додатків, паблік спікінг.



**Анджей Гузовський**

Архітектор інтернет-систем і розробник високого рівня. Позаду довгий шлях технологій від демосцени і ігробуду до високонавантажених телекомунікаційних систем. Цінує витончений декадентський кіберпанк, лірику сучасних технологій і персиковий колір.

## 5. Контакти

- 5.1. **УВАГА! Дедлайн продовжено! Рішення завантажуйте на сайті за адресою:** <http://uawebchallenge.com/semifinalVIII> до 18 жовтня, 08:00 (EEST). Рішення подані пізніше вказаного терміну зараховані не будуть.
- 5.2. **Вказуйте емейл реєстрації у формі.** Рішення з незареєстрованих емейлів прийняті не будуть.
- 5.3. Питання та уточнення щодо змісту завдання ви можете задати [ТУТ](#). Шукайте відповідь на ваше запитання стосовно змісту завдання [ТУТ](#).
- 5.4. Судді даватимуть відповіді на найбільш важливі для вирішення завдання питання та ігноруватимуть:
  - питання, які не стосуються завдання
  - питання, відповіді на які вже надані або присутні у тексті завдання.
- 5.5. Оголошення фіналістів відбудеться 10 листопада.

### Генеральні партнери сезону

