

## ТЕСТОВЕ ЗАВДАННЯ “CAR SHOWROOM”

Пропонуємо разом із нами розробити інтерфейс віртуального автосалону, який продемонструє своїм відвідувачам наявний асортимент автомобілів та дозволить висловити свої думки щодо певних моделей у коментарях.

### Звідки взяти дані?

Для виконання цього завдання скористуємось безкоштовним API [DummyJSON](#): із розділу [Products](#) можна отримати перелік різноманітних товарів, відфільтрувати їх по категорії [vehicle](#) — це і будуть наші автомобілі. В об’єкті кожного автомобіля є поле `reviews` з масивом коментарів-оглядів.

За бажанням, підключайте будь-які інші публічно доступні сервіси для генерації даних.

### На що звернути увагу?

Нижче вказані вимоги до реалізації, але вони лише задають загальні рамки. Остаточний результат залежить від вашого бачення, розуміння принципів побудови інтерфейсів та зручності їх використання. Ми не заохочуємо вас вигадувати складний дизайн — навпаки, можна обійтись стандартними елементами, або набором базових елементів з вашої улюбленої бібліотеки. Головне, щоби інтерфейс працював без помилок та виглядав гармонійно — це ми будемо оцінювати як користувачі з долею суб’єктивності. Але, даруйте, без цього ніяк.

### Розробка застосунку

Для роботи з API напишемо SPA такої структури:

- Головна сторінка /  
Компоненти сторінки:
  - Список автомобілів
  - Форма пошуку або фільтрації автомобілів за параметрами
- Індивідуальна сторінка автомобіля `/vehicles/{vehicleId}`  
Компоненти сторінки:
  - Блок розширеної інформації про автомобіль
  - Список коментарів
  - Форма додавання коментаря

### ВИМОГИ ДО РЕАЛІЗАЦІЇ

- Застосунок написаний на React.
- При заповненні форм виконується мінімальна валідація даних (перевірка на наявність значення, на максимальну довжину).
- Додані через застосунок коментарі зберігаються у локальному сховищі браузера і виводяться на сторінці після її перезавантаження.

Можна застосовувати необхідні бібліотеки на свій розсуд.

## ВИМОГИ ДО ВЕРСТКИ

Верстка має бути мінімалістичною, без фреймворків, але виглядати коректно у роздільній здатності від 320 до 1440 px (елементи не мають «злипатися» або виходити за межі екрана). Необхідно дотримуватися семантики елементів.

## ВИМОГИ ДО РОЗМІЩЕННЯ

Вихідний (не стиснутий) код застосунку треба завантажити у публічний репозиторій на GitHub.

В описі репозиторію обов'язково дайте посилання на розгорнутий застосунок. Ви можете скористатись будь-яким сервісом для розгортання та хостингу: GitHub Pages<sup>ii</sup>, Vercel, Netlify або іншим. Головне, щоби застосунок залишався доступним щонайменше тиждень з моменту надсилання вами роботи.

Акуратний, легкий для сприйняття вихідний код не менш важливий, ніж відсутність помилок у роботі застосунку.

---

<sup>ii</sup> Зважайте на обмеження при використанні клієнтського роутингу на GitHub Pages:  
<https://create-react-app.dev/docs/deployment/#notes-on-client-side-routing>