



Домашнє завдання

Компоненти React: Stateful проти Stateless

Складність: Просте

Мета: Ознайомитися з основними концепціями розробки компонентів у React, зокрема розуміння різниці між stateful (становими) та stateless (безстановими) компонентами. Навчитися створювати та використовувати обидва типи компонентів, а також зрозуміти їхні особливості та сфери застосування.

Постановка задачі:

1. Ініціалізація проекту:

- Створіть новий React проект за допомогою Vite:

```
npm create vite@latest my-react-app -- --template react
```

- Перейдіть до каталогу проекту:

```
cd my-react-app
```

- Встановіть залежності:

```
npm install
```

2. Створення компонентів:

- Розробіть два компоненти у вашому React додатку:
 - Stateful компонент:** Використовуйте хук `useState` для додавання та управління станом.

- **Stateless компонент:** Зосередьтеся на прийомі пропсів та їх відображенні.
- Виберіть відповідні назви для компонентів, щоб було очевидно, який з них є stateful, а який — stateless.

3. Тестування проекту:

- Запустіть проект:

```
npm run dev
```

- Перевірте роботу компонента у браузері.

4. Деплоймент:

- Розгорніть додаток на платформі хостингу (наприклад, Vercel або Netlify).

Додаткові завдання (необов'язково):

- Створіть компонент на основі класів, який відтворює функціональність одного з раніше створених компонентів.
- Порівняйте підходи до створення компонентів на класах та функціях, звертаючи увагу на такі аспекти:
 - Зручність розробки.
 - Читабельність коду.
 - Використання життєвого циклу компонента.

Здача завдання:

1. Репозиторій проекту:

- Завантажте код вашого проекту на GitHub або іншу платформу для зберігання коду.
- Надайте посилання на цей репозиторій.

2. Демо проекту:

- Надajte посилання на розгорнуту версію проекту на одній з платформ хостингу (Vercel, Netlify).

3. README.md:

- Додайте файл `README.md` у репозиторій з описом проекту. Він повинен містити:
 - Короткий опис проекту.
 - Інструкції зі встановлення та запуску.
 - Посилання на демо-версію.
-

Критерії оцінювання:

1. Коректність виконання:

- Проект ініціалізовано за допомогою Vite.
- Компоненти створено та використано згідно з вимогами.
- Проект запускається та працює без помилок.

2. Чистота коду:

- Код дотримується стандартів написання (наприклад, використання `const` замість `let`, правильне іменування змінних).
- Компоненти розділені на окремі файли.

3. Документація:

- Наявний файл `README.md` з описом проекту та інструкціями.

4. Демо-версія:

- Проект розгорнуто на одній з платформ хостингу (Vercel, Netlify).
-

Поради:

- Використовуйте **хуки** (`useState`, `useEffect`) для управління станом та життєвим циклом у функціональних компонентах. Це сучасний підхід, який робить код простішим і зрозумілішим.

- Розділяйте компоненти на **stateful** (зі станом) та **stateless** (без стану). Stateful компоненти використовуйте для логіки, а stateless — для відображення даних.
 - Використовуйте **деструктуризацію** для роботи з пропсами, щоб зробити код більш читабельним.
-

Приклад структури проекту:

```
my-react-app/
├── src/
│   ├── components/
│   │   ├── StatefulComponent.jsx
│   │   ├── StatelessComponent.jsx
│   │   └── ClassComponent.jsx (опціонально)
│   ├── App.jsx
│   ├── main.jsx
│   └── index.css
├── public/
├── package.json
├── README.md
└── ...
```