

# Критерії приймання

Використовуй цей [шаблон React-проекту](#) як стартову точку своєї програми.

- Створені репозиторії `goit-react-hw-03-phonebook` і `goit-react-hw-03-image-finder`
- При здачі домашньої роботи є посилання: на вихідні файли та робочу сторінку проектів на `GitHub Pages`.
- У стані компонентів зберігається мінімально необхідний набір даних, решта обчислюється
- Під час запуску коду завдання в консолі відсутні помилки та попередження.
- Для кожного компонента є окрема папка з файлом React-компонента та файлом стилів
- Все, що компонент очікує у вигляді пропсів, передається йому під час виклику.
- Імена компонентів зрозумілі та описові
- JS-код чистий і зрозумілий, використовується `Prettier`.
- Стилзація виконана `CSS-модулями` або `Styled Components`.

## 1 - Книга контактів

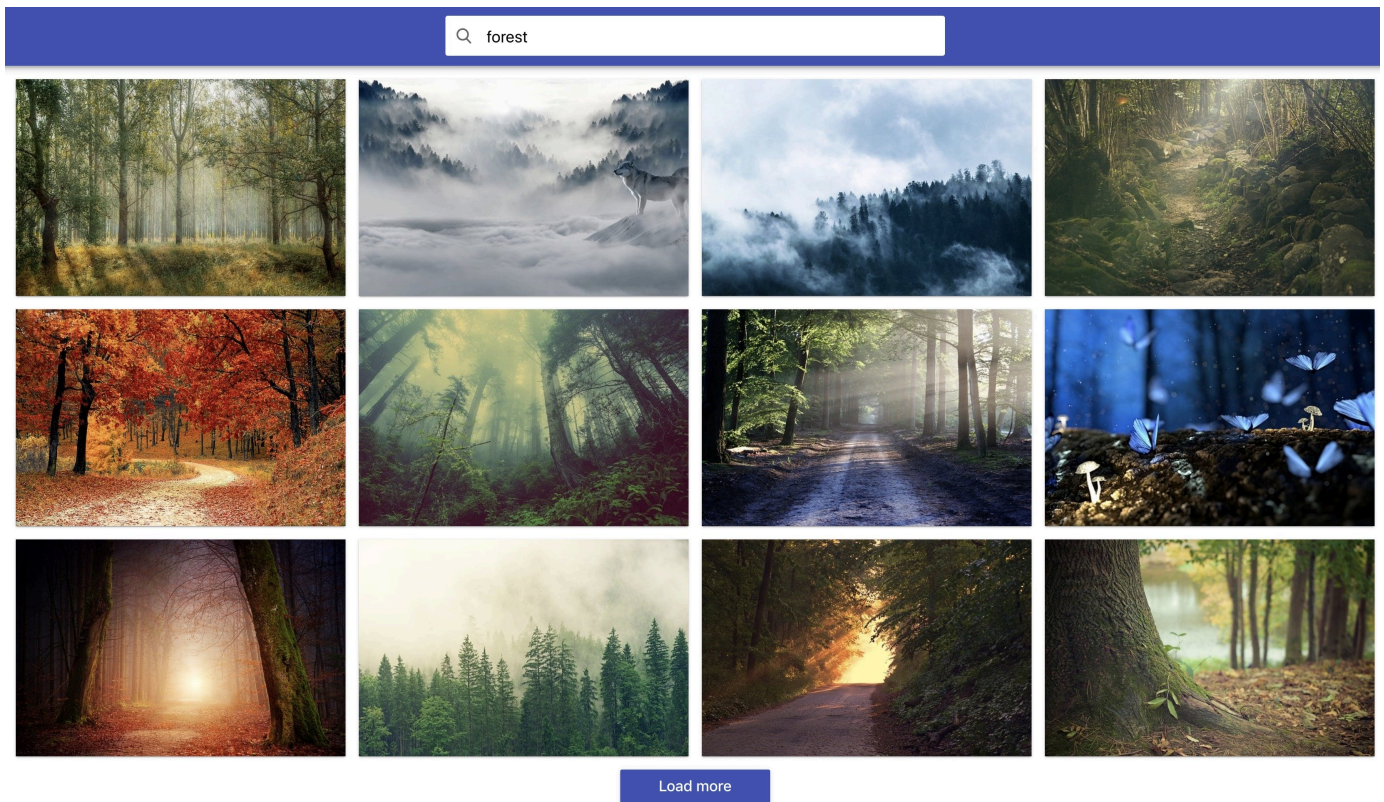
Візьми своє рішення завдання з попередньої домашньої роботи і додай зберігання контактів телефонної книги в `localStorage`. Використовуй методи життєвого циклу.

- Під час додавання та видалення контакту контакти зберігаються у локальне сховище.
- Під час завантаження застосунку контакти, якщо такі є, зчитуються з локального сховища і записуються у стан.

## 2 - Пошук зображень

Напиши застосунок пошуку зображень за ключовим словом. Прев'ю робочого застосунку [дивись за посиланням](#).

Створи компоненти `<Searchbar>`, `<ImageGallery>`, `<ImageGalleryItem>`, `<Loader>`, `<Button>` і `<Modal>`. Готові стилі компонентів можна взяти у файлі `styles.css` і підправити під себе, якщо потрібно.



## Інструкція Pixabay API

Зареєструйся та отримай приватний ключ доступу. Для HTTP-запитів використовуй публічний сервіс пошуку зображень [Pixabay](#).

URL-рядок HTTP-запиту.

```
https://pixabay.com/api/?  
q=cat&page=1&key=your_key&image_type=photo&orientation=horizontal&per_page=12
```

Pixabay API підтримує пагінацію, за замовчуванням параметр `page` дорівнює `1`. Нехай у відповіді надходить по 12 об'єктів, встановлено в параметрі `per_page`. Не забудь, що під час пошуку за новим ключовим словом, необхідно скидати значення `page` до `1`.

У відповіді від API приходить масив об'єктів, в яких тобі цікаві лише наступні властивості.

- `id` - унікальний ідентифікатор
- `webformatURL` - посилання на маленьке зображення для списку карток
- `largeImageURL` - посилання на велике зображення для модального вікна

## Опис компонента <Searchbar>

Компонент приймає один проп `onSubmit` – функцію для передачі значення інпуту під час сабміту форми. Створює DOM-елемент наступної структури.

```
<header class="searchbar">
  <form class="form">
    <button type="submit" class="button">
      <span class="button-label">Search</span>
    </button>

    <input
      class="input"
      type="text"
      autocomplete="off"
      autofocus
      placeholder="Search images and photos"
    />
  </form>
</header>
```

## Опис компонента `<ImageGallery>`

Список карток зображень. Створює DOM-елемент наступної структури.

```
<ul class="gallery">
  <!-- Набір <li> із зображеннями -->
</ul>
```

## Опис компонента `<ImageGalleryItem>`

Компонент елемента списку із зображенням. Створює DOM-елемент наступної структури.

```
<li class="gallery-item">
  <img src="" alt="" />
</li>
```

## Опис компонента `<Button>`

При натисканні на кнопку `Load more` повинна довантажуватись наступна порція зображень і рендеритися разом із попередніми. Кнопка повинна рендеритися лише тоді, коли є якісь

завантажені зображення. Якщо масив зображень порожній, кнопка не рендериться.

## Опис компонента <Loader>

Компонент спінера відображається, доки відбувається завантаження зображень.

Використовуйте будь-який готовий компонент, наприклад [react-loader-spinner](#) або будь-який інший.

## Опис компонента <Modal>

Під час кліку на елемент галереї повинно відкриватися модальне вікно з темним оверлеєм і відображатися велика версія зображення. Модальне вікно повинно закриватися по натисканню клавіші `ESC` або по кліку на оверлеї.

Зовнішній вигляд схожий на функціонал цього [VanillaJS-плагіна](#), тільки замість білого модального вікна рендериться зображення (у прикладі натисніть `Run`). Анімацію робити не потрібно!

```
<div class="overlay">
  <div class="modal">
    <img src="" alt="" />
  </div>
</div>
```