**Критерии приема**

Используй этот [шаблон React-проекта](https://github.com/goitacademy/react-homework-template#readme) как стартовую точку своего приложения.

* Созданы репозитории goit-react-hw-03-phonebook и goit-react-hw-03-image-finder
* При сдаче домашней работы есть ссылки: на исходные файлы и рабочую страницу проектов на GitHub Pages
* В состоянии компонентов хранится минимально необходимый набор данных, остальное вычисляется
* При запуске кода задания, в консоли нет ошибок и предупреждений
* Для каждого компонента есть отдельная папка с файлом React-компонента и файлом стилей
* Все что компонент ожидает в виде пропов, передается ему при вызове
* Имена компонентов понятные, описательные
* JS-код чистый и понятный, используется Prettier
* Стилизация выполнена CSS-модулями или Styled Components.

**1 - Книга контактов**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-03/#1---%D0%BA%D0%BD%D0%B8%D0%B3%D0%B0-%D0%BA%D0%BE%D0%BD%D1%82%D0%B0%D0%BA%D1%82%D0%BE%D0%B2)

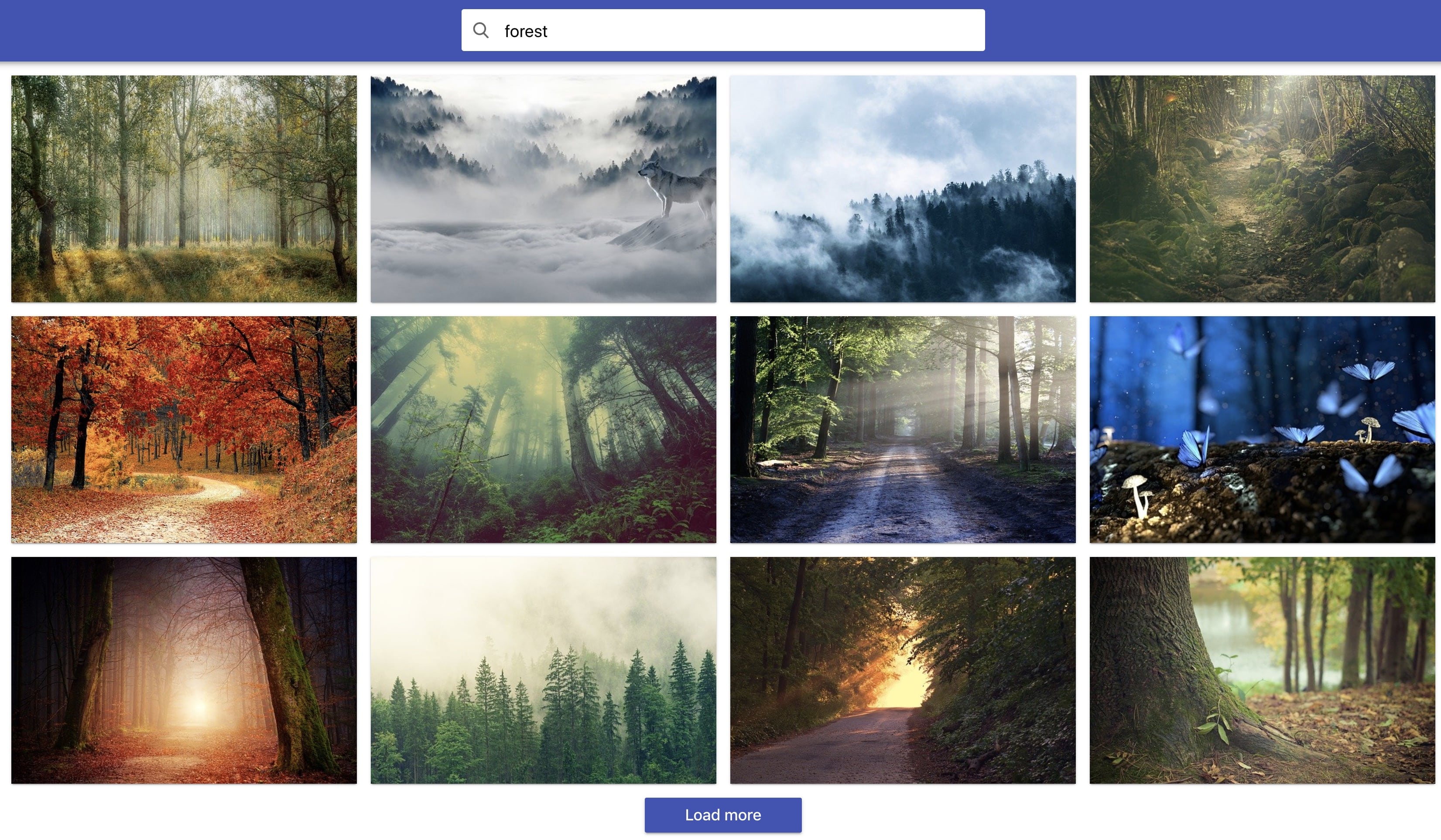
Возьми свое решение задания из предыдущей домашней работы и добавь хранение контактов телефонной книги в localStorage. Используй методы жизненного цикла.

* При добавлении и удалении контакта, контакты сохраняются в локальное хранилище.
* При загрузке приложения, контакты, если таковые есть, считываются из локального хранилища и записываются в состояние.

**2 - Поиск изображений**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-03/#2---%D0%BF%D0%BE%D0%B8%D1%81%D0%BA-%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9)

Напиши приложение поиска изображений по ключевому слову. Превью рабочего приложения [смотри по ссылке](https://drive.google.com/file/d/1oXCGyiq4uKwW0zzraZLKk4lh3voBlBzZ/view?usp=sharing).

Создай компоненты <Searchbar>, <ImageGallery>, <ImageGalleryItem>, <Loader>, <Button> и <Modal>. Готовые стили компонентов можно взять в файле [styles.css](https://minhaskamal.github.io/DownGit/#/home?url=https://github.com/goitacademy/react-homework/blob/master/homework-03/image-finder/styles.css) и подправить под себя, если необходимо.



**Инструкция Pixabay API**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-03/#%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D0%B8%D1%8F-pixabay-api)

Для HTTP-запросов используй публичный сервис поиска изображений [Pixabay](https://pixabay.com/api/docs/" \t "_blank). Зарегистрируйся и получи приватный ключ доступа.

URL-строка HTTP-запроса.

https://pixabay.com/api/?q=cat&page=1&key=your\_key&image\_type=photo&orientation=horizontal&per\_page=12

Pixabay API поддерживает пагинацию, по умолчанию параметр page равен 1. Пусть в ответе приходит по 12 объектов, установлено в параметре per\_page. Не забудь что при поиске по новому ключевому слову, необходимо сбрасывать значение page в 1.

В ответе от апи приходит массив объектов, в которых тебе интересны только следущие свойства.

* id - уникальный идентификатор
* webformatURL - ссылка на маленькое изображение для списка карточек
* largeImageURL - ссылка на большое изображение для модального окна

**Описание компонента <Searchbar>**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-03/#%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%B0-searchbar)

Компонент принимает один проп onSubmit - функцию для передачи значения инпута при сабмите формы. Создает DOM-элемент следующей структуры.

<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>

**Описание компонента <ImageGallery>**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-03/#%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%B0-imagegallery)

Список карточек изображений. Создает DOM-элемент следующей структуры.

<ul class="gallery">  
 *<!-- Набор <li> с изображениями -->*  
</ul>

**Описание компонента <ImageGalleryItem>**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-03/#%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%B0-imagegalleryitem)

Компонент элемента списка с изображением. Создает DOM-элемент следующей структуры.

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

**Описание компонента <Button>**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-03/#%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%B0-button)

При нажатии на кнопку Load more должна догружаться следующая порция изображений и рендериться вместе с предыдущими. Кнопка должна рендерится только тогда, когда есть какие-то загруженные изобаржения. Если массив изображений пуст, кнопка не рендерится.

**Описание компонента <Loader>**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-03/#%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%B0-loader)

Компонент спинера, отображется пока идет загрузка изобаржений. Используй любой готовый компонент, например [react-loader-spinner](https://github.com/mhnpd/react-loader-spinner) или любой другой.

**Описание компонента <Modal>**[**​**](https://textbook.edu.goit.global/lms-react-homework/v1/docs/hw-03/#%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%B0-modal)

При клике по элементу галереи должно открываться модальное окно с темным оверлеем и отображаться большая версия изображения. Модальное окно должно закрываться по нажатию клавиши ESC или по клику на оверлее.

Внешний вид похож на функционал этого [VanillaJS-плагина](https://basiclightbox.electerious.com/" \t "_blank), только вместо белого модального окна рендерится изображение (в примере нажми Run). Анимацию делать не нужно!

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