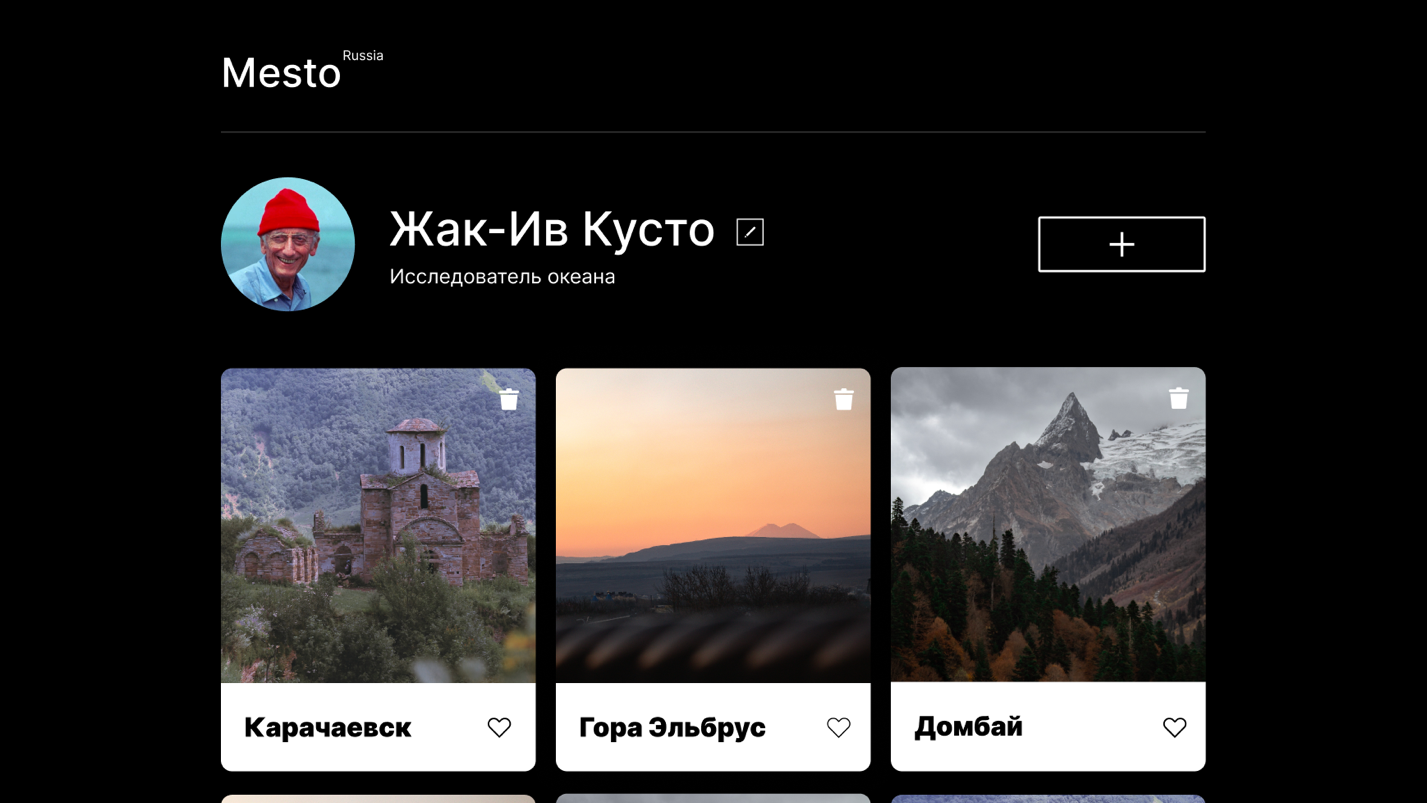
**Проектная работа 5**

В этой проектной работе вы продолжите разрабатывать функциональность проекта Mesto:



*Так будет выглядеть готовая проектная работа.*

**0. Сдача проектной работы**

Продолжайте писать код в проекте, который сделали за предыдущие две недели. Тогда по истории коммитов будет видно, что вы создали этот проект от начала и до конца.

Готовую работу не забудьте запушить к себе в «Гитхаб». Репозиторий тот же — mesto.

**1. Шесть карточек «из коробки»**

При загрузке на странице должно быть 6 карточек, которые добавит JavaScript. Их названия и фотографии выберите сами или возьмите готовый массив:

Скопировать кодJAVASCRIPT

const initialCards = [

{

name: 'Архыз',

link: 'https://pictures.s3.yandex.net/frontend-developer/cards-compressed/arkhyz.jpg'

},

{

name: 'Челябинская область',

link: 'https://pictures.s3.yandex.net/frontend-developer/cards-compressed/chelyabinsk-oblast.jpg'

},

{

name: 'Иваново',

link: 'https://pictures.s3.yandex.net/frontend-developer/cards-compressed/ivanovo.jpg'

},

{

name: 'Камчатка',

link: 'https://pictures.s3.yandex.net/frontend-developer/cards-compressed/kamchatka.jpg'

},

{

name: 'Холмогорский район',

link: 'https://pictures.s3.yandex.net/frontend-developer/cards-compressed/kholmogorsky-rayon.jpg'

},

{

name: 'Байкал',

link: 'https://pictures.s3.yandex.net/frontend-developer/cards-compressed/baikal.jpg'

}

];

[Фотографии можно взять с сайта Unsplash](https://unsplash.com/) — коллекции бесплатных изображений. Тогда не придётся думать об авторских правах.

**2. Форма добавления карточки**

Добавьте в проект форму добавления новой карточки. [Макет найдёте в «Фигме»](https://www.figma.com/file/bjyvbKKJN2naO0ucURl2Z0/JavaScript.-Sprint-5?node-id=0%3A1).

Сделайте так, чтобы форма открывалась нажатием на кнопку «+» и закрывалась кликом на крестик:

Так открывается и закрывается форма добавления карточек.

**3. Добавление карточки**

Дайте пользователю возможность добавлять карточки:

Можно написать имя карточки и дать ссылку на картинку.

Сделайте так, чтобы при клике на «сохранить» новая карточка попадала в начало контейнера с ними. А диалоговое окно после добавления автоматически закрывалось.

Чтобы создавать новые карточки, добавьте обработчик событий submit. Сделайте это аналогично прошлому спринту, в котором вы настраивали редактирование информации о пользователе.

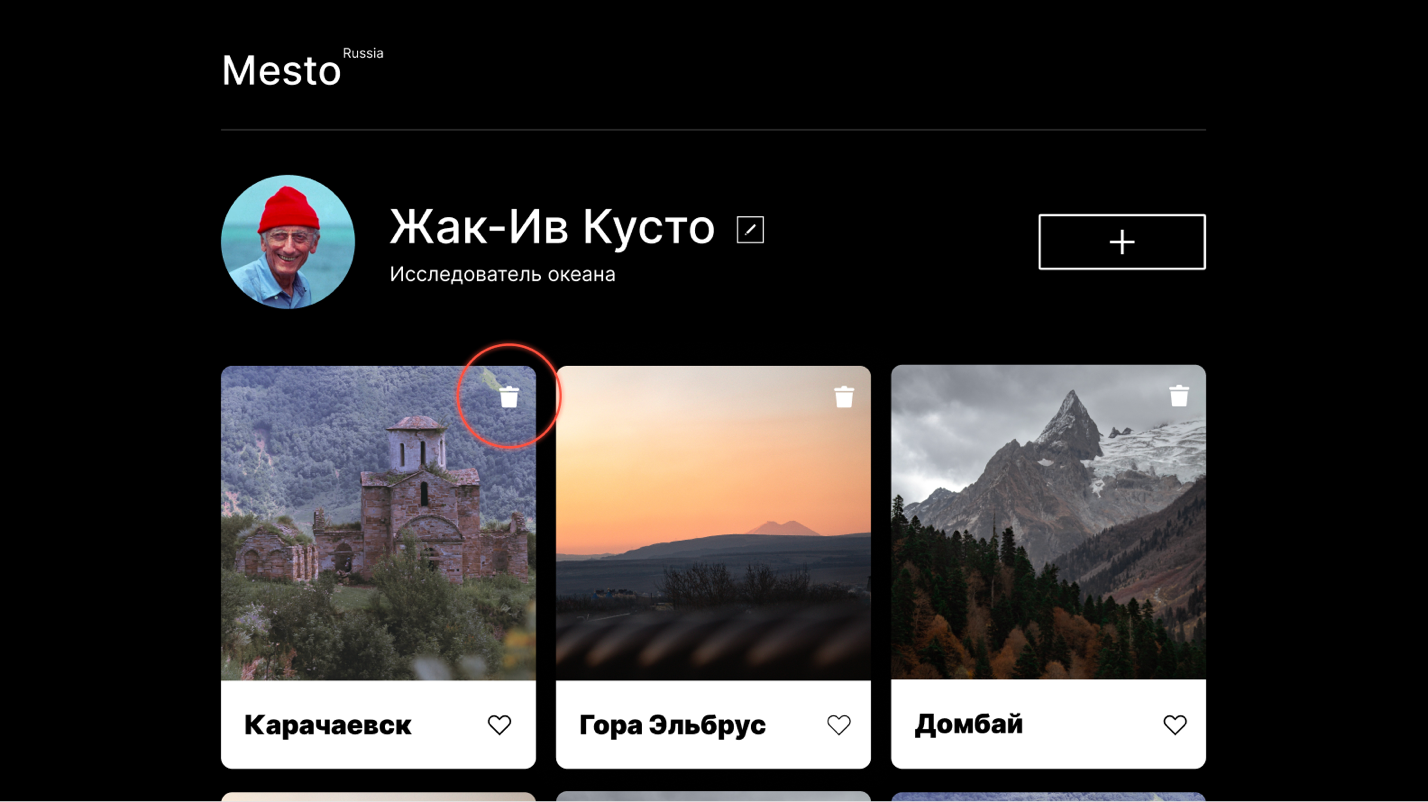
**4. Лайк карточки**

Сделайте так, чтобы карточки можно было лайкать:

Если лайкнуть карточку, сердечко поменяет цвет.

**5. Удаление карточки**

Добавьте карточкам иконку удаления. [Должно выглядеть как в макете](https://www.figma.com/file/bjyvbKKJN2naO0ucURl2Z0/JavaScript.-Sprint-5?node-id=0%3A1):



*Кнопка «удалить» — классическая урна*

Теперь настройте, чтобы карточка удалялась при клике на эту иконку:

Пришлось пожертвовать Архызом.

**6. Открытие попапа с картинкой**

Настройте просмотр фотографий. Пусть открываются нажатием на картинку и закрываются кликом на крестик:

Рассмотрим Челябинскую область поближе.

Требования к вёрстке:

* [внешний вид попапа с картинкой как на макете](https://www.figma.com/file/bjyvbKKJN2naO0ucURl2Z0/JavaScript.-Sprint-5?node-id=0%3A1);
* файловая структура организована по БЭМу;
* вёрстка адаптивная, без полосы прокрутки при изменении ширины окна. Максимальная ширина открытой картинки — 75vw, максимальная высота — 75vh.

**7. Плавное открытие и закрытие попапов**

Сделайте так, чтобы все попапы плавно открывались. Пусть проявляются из прозрачности и уходят в неё при закрытии:

Свойство display:none не позволит сделать плавное переключение между состояниями карточек. Нужно что-то придумать.

**Подсказка.** Если зашли в тупик, прочитайте [вот этот тред](https://stackoverflow.com/questions/3331353/transitions-on-the-css-display-property) на StackOverflow. Там в первом ответе вы найдёте два CSS-свойства, которые помогут.

**Заключение**

Если перезагрузить страницу, карточки и лайки не сохранятся. Для этого пришлось бы писать бэкэнд. Тогда бы сайт получал карточки с сервера и там же сохранял новые. Вы же в этом спринте делали только фронтенд сайта, логику его интерактивности.

В реальных проектах так и происходит: фронтенд и бэкенд пишутся отдельно, а потом их связывают друг с другом.