Техническое задание

О проекте

Кекстаграм — сервис просмотра изображений. Пользователям предоставлена возможность загружать свои фотографии или просматривать фотографии, загруженные ранее другими пользователями.

Описание функциональности

1. Загрузка нового изображения на сайт и заполнение информации о нём:

1.1. Загрузка нового изображения:

выбор файла с изображением для загрузки;

изменение масштаба изображения;

применение одного из заранее заготовленных эффектов;

выбор глубины эффекта с помощью ползунка;

добавление текстового комментария;

добавление хэш-тегов.

Обратите внимание: в последующих после выбора файла шагах может использоваться тестовое изображение, вставленное в разметку по умолчанию. Подстановка в форму реального изображения, выбранного пользователем — это необязательная функциональность.

1.2. Выбор изображения для загрузки осуществляется с помощью стандартного контрола загрузки файла #upload-file, который стилизован под букву «О» в логотипе. После выбора изображения (изменения значения поля #upload-file), показывается форма редактирования изображения. У элемента .img-upload\_\_overlay удаляется класс hidden, а body задаётся класс modal-open.

1.3 Закрытие формы редактирования изображения производится либо нажатием на кнопку #upload-cancel, либо нажатием клавиши Esc. Элементу .img-upload\_\_overlay возвращается класс hidden. У элемента body удаляется класс modal-open.

2. Редактирование изображения и ограничения, накладываемые на поля

2.1. Масштаб:

При нажатии на кнопки .scale\_\_control--smaller и .scale\_\_control--bigger должно изменяться значение поля .scale\_\_control--value;

Значение должно изменяться с шагом в 25. Например, если значение поля установлено в 50%, после нажатия на «+», значение должно стать равным 75%. Максимальное значение — 100%, минимальное — 25%. Значение по умолчанию — 100%;

При изменении значения поля .scale\_\_control--value изображению внутри .img-upload\_\_preview должен добавляться соответствующий стиль CSS, который с помощью трансформации scale задаёт масштаб. Например, если в поле стоит значение 75%, то в стиле изображения должно быть написано transform: scale(0.75).

2.2. Наложение эффекта на изображение:

По умолчанию должен быть выбран эффект «Оригинал».

На изображение может накладываться только один эффект.

При смене эффекта, выбором одного из значений среди радиокнопок .effects\_\_radio, добавить картинке внутри .img-upload\_\_preview CSS-класс, соответствующий эффекту. Например, если выбран эффект .effect-chrome, изображению нужно добавить класс effects\_\_preview--chrome.

Интенсивность эффекта регулируется перемещением ползунка в слайдере. Слайдер реализуется сторонней библиотекой для реализации слайдеров noUiSlider. Уровень эффекта записывается в поле .effect-level\_\_value. При изменении уровня интенсивности эффекта (предоставляется API слайдера), CSS-стили картинки внутри .img-upload\_\_preview обновляются следующим образом:

Для эффекта «Хром» — filter: grayscale(0..1) с шагом 0.1;

Для эффекта «Сепия» — filter: sepia(0..1) с шагом 0.1;

Для эффекта «Марвин» — filter: invert(0..100%) с шагом 1%;

Для эффекта «Фобос» — filter: blur(0..3px) с шагом 0.1px;

Для эффекта «Зной» — filter: brightness(1..3) с шагом 0.1;

Для эффекта «Оригинал» CSS-стили filter удаляются.

При выборе эффекта «Оригинал» слайдер скрывается.

При переключении эффектов, уровень насыщенности сбрасывается до начального значения (100%): слайдер, CSS-стиль изображения и значение поля должны обновляться.

2.3. Хэш-теги:

хэш-тег начинается с символа # (решётка);

строка после решётки должна состоять из букв и чисел и не может содержать пробелы, спецсимволы (#, @, $ и т. п.), символы пунктуации (тире, дефис, запятая и т. п.), эмодзи и т. д.;

хеш-тег не может состоять только из одной решётки;

максимальная длина одного хэш-тега 20 символов, включая решётку;

хэш-теги нечувствительны к регистру: #ХэшТег и #хэштег считаются одним и тем же тегом;

хэш-теги разделяются пробелами;

один и тот же хэш-тег не может быть использован дважды;

нельзя указать больше пяти хэш-тегов;

хэш-теги необязательны;

если фокус находится в поле ввода хэш-тега, нажатие на Esc не должно приводить к закрытию формы редактирования изображения.

Сообщения о неправильном формате хэштега задаются с помощью метода setCustomValidity у соответствующего поля.

2.4. Комментарий:

комментарий не обязателен;

длина комментария не может составлять больше 140 символов;

если фокус находится в поле ввода комментария, нажатие на Esc не должно приводить к закрытию формы редактирования изображения.

3. Отправка данных на сервер

3.1. После заполнения всех данных, при нажатии на кнопку «Отправить», все данные из формы, включая изображения, с помощью AJAX-запроса отправляются на сервер https://22.javascript.pages.academy/kekstagram методом POST с типом multipart/form-data.

3.2. Страница реагирует на неправильно введённые значения в форму. Если данные, введённые в форму, не соответствуют ограничениям, указанным в пунктах 2.3 и 2.4, форму невозможно отправить на сервер. При попытке отправить форму с неправильными данными, отправки не происходит, а неверно заполненные поля подсвечиваются красной рамкой. Способ добавления рамки и её стиль произвольные.

3.3. При успешной отправке формы форма редактирования изображения закрывается, все данные, введённые в форму, и контрол фильтра приходят в исходное состояние:

масштаб возвращается к 100%;

эффект сбрасывается на «Оригинал»;

поля для ввода хэш-тегов и комментария очищаются;

поле загрузки фотографии, стилизованное под букву «О» в логотипе, очищается.

3.4. Если отправка данных прошла успешно, показывается соответствующее сообщение. Разметку сообщения, которая находится блоке #success внутри шаблона template, нужно разместить в <main>. Сообщение должно исчезать после нажатия на кнопку .success\_\_button, по нажатию на клавишу Esc и по клику на произвольную область экрана за пределами блока с сообщением.

3.5. Если при отправке данных произошла ошибка запроса, нужно показать соответствующее сообщение. Разметку сообщения, которая находится блоке #error внутри шаблона template, нужно разместить в <main>. Сообщение должно исчезать после нажатия на кнопки .error\_\_button, по нажатию на клавишу Esc и по клику на произвольную область экрана за пределами блока с сообщением.

3.6. Нажатие на кнопку #upload-cancel приводит к закрытию формы и возвращению всех данных и контрола фильтра к исходному состоянию (описано в пункте 3.3). Поле загрузки фотографии, стилизованное под букву «О» в логотипе, очищается.

4. Просмотр загруженных изображений

4.1. Загрузка изображений от других пользователей производится сразу после открытия страницы с удалённого сервера: https://22.javascript.pages.academy/kekstagram/data.

4.2. Если при загрузке данных с сервера произошла ошибка запроса, нужно показать соответствующее сообщение. Дизайн блока с сообщением нужно придумать самостоятельно.

4.3. Все загруженные изображения показаны на главной странице в виде миниатюр. DOM-элемент миниатюры генерируется на основе шаблонного элемента picture, расположенного в элементе template на странице.

4.4. При нажатии на любую из миниатюр, показывается блок .big-picture, содержащий полноэкранное изображение с количеством лайков и комментариев. Элементу body задаётся класс modal-open. Данные, описывающие изображение, должны подставляться в соответствующие элементы в разметке.

4.5. Выход из полноэкранного режима просмотра фотографии осуществляется либо нажатием на иконку крестика .big-picture\_\_cancel в правом верхнем углу блока .big-picture, либо нажатием на клавишу Esc. У элемента body удаляется класс modal-open.

4.6. Все комментарии к изображению выводятся в блок .social\_\_comments. Сразу после открытия изображения в полноэкранном режиме отображается не более 5 комментариев. Пример разметки списка комментариев приведён в блоке .social\_\_comments. Комментарий оформляется отдельным элементом списка li с классом social\_\_comment. Аватарка автора комментария отображается в блоке .social\_\_picture. Имя автора комментария отображается в атрибуте alt его аватарки. Текст комментария выводится в блоке .social\_\_text.

4.7. Отображение дополнительных комментариев происходит при нажатии на кнопку .comments-loader. При нажатии на кнопку отображается не более 5 новых комментариев.

4.8. Если все комментарии показаны, кнопку .comments-loader следует скрыть, добавив класс hidden.

5. Фильтрация изображений от других пользователей

5.1. Доступные фильтры:

«По умолчанию» — фотографии в изначальном порядке с сервера;

«Случайные» — 10 случайных, не повторяющихся фотографий;

«Обсуждаемые» — фотографии, отсортированные в порядке убывания количества комментариев.

5.2. Блок, с помощью которого производится фильтрация фотографий, скрыт изначально и показывается только после окончания загрузки всех фотографий.

5.3. При переключении фильтров, отрисовка изображений, подходящих под новый фильтр, должна производиться не чаще, чем один раз 500 мс (устранение дребезга).

6. Необязательная функциональность

6.1. После выбора изображения пользователем с помощью стандартного контрола загрузки файла #upload-file, нужно подставить его в форму редактирования вместо тестового изображения.

6.2. В проекте настроена сборка JavaScript с помощью Webpack.