Upload photo
Optional

Add text

Add link
Optional

POST

This is the default view. The user has not entered any information.

There are three fields. 1 is required and 2 are optional.

Image - optional. The image can be any size and dimension. It must be a JPEG or PNG. Users should be able to click in the gray photo box (to open file browser) or drag a photo into the photo box.

Text - required. 60 character maximum.

Link - optional. To ensure consistency, we need to strip http:// off the link on save if entered by the user.

In the Model.py file, you'll see that the form needs to also save the User and Timestamp automatically. The included Views.py file already accomplishes this. Although, we would like it upgraded to AJAX so results can be returned with JSON without refreshing the page. This is not accomplished yet for this form but we did include a sample of another form from the website where this is accomplished (in the Views.py file)

Notes:
The dark background in the mockup is not significant. It is just included in the mockup to show the form box clearly. It is not implying that the form should be in a lightbox. It should not be.

Add text

Add link
Optional

CANCEL   POST
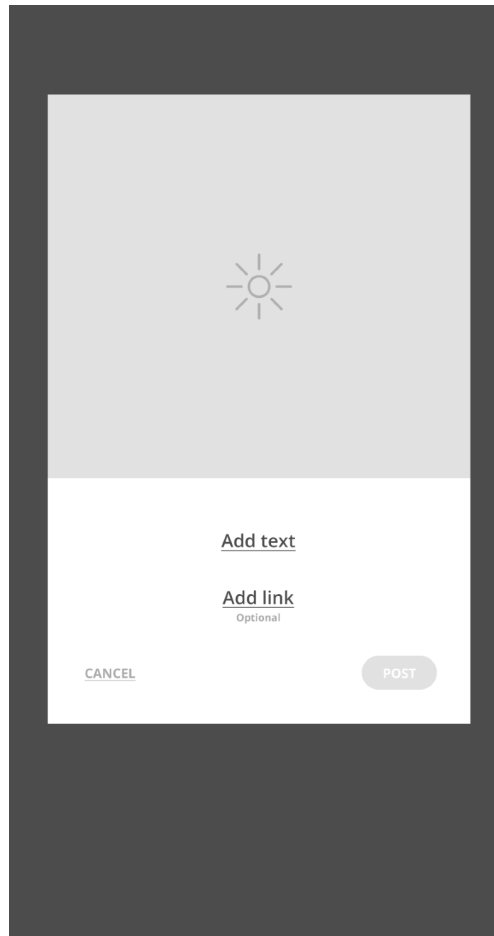
While an image is loading, a loading graphic should display on the screen. This is represented by the graphic in the center of the gray photo box in this mockup. Please use that image for the placeholder. We will replace it with an animated GIF at a later time.

This is the view after the user has entered content but has to yet submitted.

The photo should fill the width of the box with a variable height depending on the dimension of the image they uploaded.

When entering text, the gray bubble appears to countdown the amount of available characters left.

Once text has been entered, the cancel button appears and the post button turns green. The cancel button will close this screen.

A link that is longer than the width of this box will overflow like a typical input box and hide the first characters.
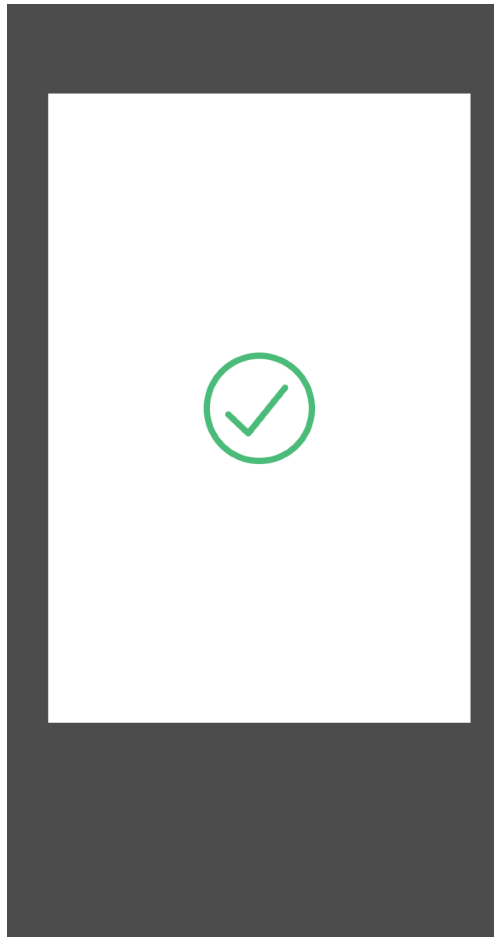
Hovering over an uploaded photo will give the user an option to delete the photo by clicking on the X. Doing so, will return the photo box to the default view.

After the form is successfully submitted, a single checkmark image should be returned and displayed.

Notes:

When a form is successfully submitted, it should have uploaded the image to Cloudinary (not to the local server) and saved the text content in the local server database (with a reference to the Cloudinary image).