Picopy

Milestone 2

Jett Crowson, Benton Haensel, Jacob Dewild

Justin Floyd, Sarah Bordiuk, Will Huynh

Picopy Feature List

Upload an image

Picopy will feature the ability to upload an image from the users local machine into our server.

Filters

There will be a selection of filters that can easily be applied to the users photo.

At the time of the writeup, Picopy is set to include:

- Grayscale
- Distortion
- Pixel art
- Borders
- Color filters

Though these may be subject to change as development progresses.

Save a photo locally

User edited photos will be able to be saved to the users local machine with ease.

Picopy Gallery

The Picopy Gallery will be a cloud based collection of photos from all Picopy users. View creations from around the world. No login needed.

Save your editing session

With Picopy, it’s easy to step away from your computer and return to your editing session with ease. Users can save their state and return whenever they want.

Social media integration

Like your brand new edited photo? Share it with your followers! With Picopy it's easy to post your creation to your social media.

Picopy Requirements

Functional and Non-functional Requirements

Upload an image

Functional:

Users will be able to upload a photo they would like to edit.

Non-Functional:

- Photo will be uploaded from a local directory through an html form
- Photo will be loaded onto an html canvas to be further manipulated

Add a filter to a photo

Functional:

- A user can select a filter to be applied to the uploaded photo
- A user can select options for the filters
- A user can select multiple filters

Non-Functional:

- Filter can be applied to photo by selecting corresponding radio button and clicking go
- CSS filters will be able to be applied
- Custom filters will be implemented through relevant javascript photo editing libraries
- Picopy will use CSS or javascript algorithms to manipulate the photo through the html canvas
- Filters will have options associated with them (e.g. the level of pixelation)

Save an edited photo

Functional:

Users will be able to click a button to save their image locally

User will be able to click a seperate, Dropbox button to save their photo to the cloud

Non-Functional:

- The button will prompt the user for which directory to save the photo to, implemented through relevant html and javascript methods.
- Cloud based storage will implemented through Dropbox's “JavaScript Saver API”
- User will be prompted to log into dropbox and will be able to save their photo to their personal Dropbox

User will be able to view/use a gallery

Functional:

User has the option the add their photo to the Picopy Gallery
User can navigate to the gallery to see a showcase of other picopy photos
Any user on the site can view the gallery.
Non-Functional:

- Picopy photos are pushed to database when the option is selected
- When the user enters the gallery, photos will be systematically loaded into the gallery from the database
- Database will be implemented through node.js and stored on the server.
- Photos will be showcased in a simple, minimalist grid, implemented with css (https://css-tricks.com/seamless-responsive-photo-grid/)

Resume an editing session

Functional:
Users will be able to resume previous picopy sessions after leaving the session.
Non-Functional:

- Picopy will have a save state for each user in order to resume the previous editing, implemented through Amazon S3

Social Media Integration

Functional:
User has the ability to post their edited photo to Twitter or Facebook
Non-Functional:

- Twitter integration to be implemented through Twitter's social media integration api
- Facebook integration to be implemented through Facebook's social media integration api

Project Plan:

https://app.asana.com/share/colorado/picopy/221372384534556/ded04b06a1c76dfe574d7997460f6d45

Must have @colorado.edu email to join/view

Group members are able to select what part of the project they want to develop