Assignment 2: Chrome Extension for Web Instagram

TA-in-charge: Yi Wang
yiwang@cse.cuhk.edu.hk

1 Introduction

In the first assignment, you have implemented the back-end component of a Web Instagram application, which allows multiple users to host their images online. However, the Web Instagram application is a toy application that supports limited functionalities, *e.g.*, uploading images and applying filters. You will enhance the functionality of the Web Instagram through client side techniques in the second assignment.

Specifically, you will implement a Chrome extension. With the extension, a user will be able to perform advanced editing on her/his images. For example, the user will be able to adjust the brightness, contrast, saturation, *etc.*, of an image, and to apply advanced preset filters. Further, it enables a user to save any image she/he likes to her/his album from any website she/he is browsing.

You will not implement the extension for your own Web Instagram implementation, which may differ significantly from others' implementations. Instead, you will implement the extension for an open-source image hosting application – Chevereto[1]. The features of your Chrome extensions are as follows. Specifically, your extension should

- support Chevereto hosted on any server;
- enhance the photo editing feature of Chevereto using CamanJS[2];
- allow a user to upload web image(s) with right click(s).

1.1 Restrictions

There are only a few restrictions in this assignment.

- You have to use the CamanJS image editing JavaScript library.
- You can use any extra libraries (*e.g.*, jQuery) besides CamanJS in this assignment. (I know this is not a restriction; this point is to tell you the removal of all previous restrictions.)
- You must implement a Google Chrome Extension.

[1] https://github.com/Chevereto/Chevereto-Free/
[2] http://camanjs.com/

2 Implementation, Grading, and Constraints

- On the extension side, you are welcome to write UGLY interfaces on the popup page. Remember, we grade your work based on the completion of features, not how beautiful your interface is.
- To submit your work, compress your codes and documents into one single file named as 'name_id.zip', and then submit it through blackboard.
- Your extension should be able to run on top of Google Chrome 60.0 or above.
- We will be using Windows 7/10 or Mac OS X as the desktop client platform during the demonstrations.
- There must not be any JavaScript error thrown to the Developer Console from your extension (from the background, popup, and content scripts). Otherwise, at least 5 points would be deducted.

Task 1: Setting up Chevereto in Your Local Machine

Chevereto is an 'Instagram-like' open-source image hosting application. In this assignment, Chevereto is required to be deployed in a local machine for testing the functionality of your Chrome extension. You can find the installation instruction of Chevereto at `https://github.com/Chevereto/Chevereto-Free`.

Task 2: Extension Setting

Your Chrome extension should use the Browser Action so that a user can upload images on any website to Chevereto. Since the extension needs to interact with an instance of Chevereto, the specific host name of the Chevereto instance must be configured in the configuration file of the extension. The default value needs to be set to `http://localhost/`. The user should be able to change this default value to any other valid URL by clicking a Setting button in the Popup of the browser action of your extension. In particular, you should direct the user to an options page to configure the Chevereto URL.

Hint: The setting should be saved into the browser storage, *i.e.*, using the *chrome.storage* API.

Task 3: Enhancing Chevereto's Photo Editing Feature

When a user clicks the Upload button in the main page, Chevereto allows a user to upload images from a user's computer or from the Internet by providing image URLs in its Image Upload interface/page. It then directs the user to an Image Preview page that allows the user to edit or resize the selected images.

In this task, you are required to enhance the photo editing feature of Chevereto by leveraging CamanJS. Specifically, you need to insert a new Edit Image with CamanJS button next to each image displayed in the Image Preview page. When a user clicks on such a button, an image editor interface is displayed such that the user can edit the corresponding image with the CamanJS library. Note the image editor interface should be displayed in the same web page instead of a new page. Your image editor should provide exactly the same features demonstrated in `http://camanjs.com/examples/`, *e.g.*, the

ones in the Interactive Example and the Preset Example. In particular, you should add an Original Image button such that the user can discard one of the preset filters or any other adjustments.

When the user finishes editing, the user can click a Done button in the image editor interface to save all the work done on an image and go back to the Image Preview page. If the user clicks the Edit Image with CamanJS button again for the same image, the user should be able to continue editing from the last saved state rather than from the original image. For example, if the *Contrast* value was set to +10 in the last edit, the user should see that the *Contrast* value is set to +10 automatically. The user can also click a Cancel button in the image editor interface to discard all adjustments made in the current editing session.

Finally, the user can click the Upload button in the Image Preview page of Chevereto to finish image uploading. The adjusted images should be uploaded to Chevereto.

Task 4: Uploading Web Images with Right Click

Your extension should insert a new option – Upload to Chevereto – into the context menu when a user right clicks an image in a web page. When the user clicks this new option, the URL of the selected image is added into the upload queue of your extension. All the selected images should not be immediately uploaded to Chevereto. The user can keep adding additional images from any web page by clicking the option in the Chrome context menu.

In the Popup of your browser action, you should provide an Upload To Chevereto button (see Figure 2). By clicking this button, a user can finish uploading to Chevereto. You should also display the number of images that have been added into the upload queue (not shown in Figure 2), and implement a drop-down list that shows all the image URLs in the queue. A user should be able to toggle the drop-down list on and off with a Show Upload Queue button.

To simplify your work, your extension can open the Image Upload page of Chevereto in a new tab. Your extension then automatically fills in all the URLs in the queue, and directs the user to the Image Preview page of Chevereto. The user can then edit those images with the features implemented in Task 3.

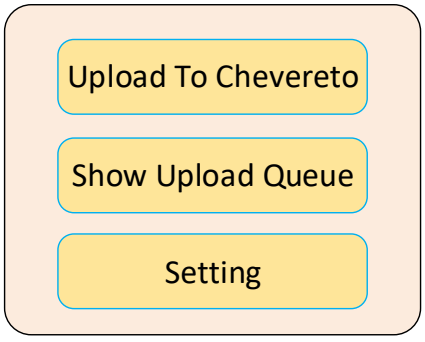

Figure 1: The popup page of the browser action.

3 Milestones

Task 1	**0%**
Task 2	**10%**
Task 3	**60%**
Injecting Edit Image with CamanJS button and showing the Photo Editor for the selected image	10%
Photo Editor - CamanJS Interactive Example	15%
Photo Editor - CamanJS Preset Example	15%
Photo Editor - Done and Cancel	5%
Photo Editor - Resume from last edit	10%
Uploading the adjusted images to Chevereto server	5%
Task 4	**30%**
Adding option in context menu	5%
Adding image URL into the upload queue	5%
Displaying information of upload queue	5%
Uploading to Chevereto	15%