Shivang Saxena
CS 410
Final Project documentation

An overview of the function of the code

The code provided is a Chrome Extension that provides a **Github Code Annotation System.** When viewing any source code file on Github, it allows the user to create annotations with user-provided text content at specific line numbers in the source code. These can be very helpful for note-taking, asking questions, or generally marking up text documents (in this case source code) with explanations and additional documentation. Whenever the user re-visits the source code file, the annotations are automatically retrieved from storage and displayed along the source code. Additionally, the annotations can be easily deleted by clicking on them if they are no longer needed.

Documentation of how the software is implemented

The annotation software is implemented as a Chrome Extension, consisting of JavaScript, HTML, and CSS. This section is broken down into modules, and how each module is implemented.

Module 1: Storage of annotations

The storage of annotations is done via Chrome's *chrome.storage.sync* API. This API allows us to store key/value pairs in the user's browser, with the added benefit that these values are automatically synced across the user's browsers if the user has multiple devices. The KEY used in storing these annotations is the URL of the source code file, which makes these annotations very easy to retrieve by simply looking up the URL the user is on. The value stored is an array containing a list of annotation JavaScript objects, each with these properties:

- **id** (Integer) - this provides a unique way to reference an annotation for deletion later
- **content** (String) - this is the text content of the annotation
- **line_number** (Integer) - this allows us to define where the annotation should appear in the source code file

Module 2: Creation of Annotations

Creating annotations should be a seamless experience with as least friction as possible. To accomplish this, a UI was created that allows specifying the annotation text content and which line it should appear on. This UI was created in HTML and JavaScript in the "popup" window that is shown when a chrome extension is clicked. When a user provides the annotation content, the *chrome.storage.sync.get* function is first called to retrieve the annotations for the current page (if any), and then the new annotation is pushed onto the annotation list, and then *chrome.storage.sync.set* is called to update the annotation list for the current page URL. Finally, the page is reloaded to allow rerendering of the annotations.

Module 3: Retrieving and Rendering of Annotations

Annotations are retrieved once the tab is finished loading in the "background" script of the chrome extension. The background script allows us to interface with chrome APIs such as *chrome.tabs.onUpdated*, which tells us when a tab is finished loading and allows us to retrieve the URL for the current tab. Using this, in conjunction with the *chrome.storage.sync.get* API, we can retrieve the current annotations for the URL (or return none if there are none).

The background script, however, cannot interface directly with the HTML DOM so we cannot directly add the annotations once they are retrieved. Using *message passing*, with the *chrome.tabs.sendMessage* API, we send the current annotations for the current url to the "content script" of the chrome extension. The content script is able to directly manipulate the DOM, which is required if we wish to add annotations to it. Once the content script receives the annotations from the background script, we can render them using the *document* API of HTML by creating an element, styling it using CSS to make it appear like a sticky-note, and injecting it into the DOM.

Documentation of the usage of the software

The extension can easily by installed following these steps:

1. Unzip the provided zip file containing the chrome extension code in the github repository (named "chrome_extension.zip here: https://github.com/Shivang44/CourseProject)
2. Click the 3 dots at the top right of any Chrome window and click "Settings"
3. On the left side click "Extensions"
4. On the top left click "Load unpacked" and navigate to the extracted source code of the extension from step #1

To use the software, simply go to any source code file in Github (for example, https://github.com/docker/cli/blob/master/cli/required.go), and then in the top right click the extension and provide the annotation text and line number you wish to make the annotation appear on. Then, click "Create". This will reload the page and the annotation will appear next to the line number you specified.

Contribution of team members

Since this was a solo project, I (Shivang Saxena) performed 100% of the tasks specified above.