

design.txt

Manifest.json file directs how the extension should run. "Browser-action" defines what should happen when user clicks the icon. Icon16.png is used for the icon, the title when user hovers over icon is "Click here to like posts!" By default, popup.html will appear. Two icons are included in "icons": icon16.png and icon128.png which are 16px by 16px and 128px by 128px respectively. The former is used for the extension and icon and the latter is used for the logo in chrome://extensions. Permissions gives the extensions permissions such as tabs which gives access to url of a tab. Content scripts injects content.js into any webpage that has prefix http: or https: .

When the user first clicks the icon, popup.html (that runs popup.js) will show a search bar and two buttons. In popup.js, we implemented code so that the user can either click "Search" or press "Enter" on the keyboard to move on to the next step. The search bar works by adding the user's input attached to the end of "https://facebook.com/search/top/?q=", so that it directs the user to a Facebook search of the name. Then, popup.js will tell the browser to automatically click the first result, which will then take the user to the friend's timeline. Pressing the search button or the enter key will take user to page2.html which has "Like" and "Go back." Pressing "Change Page" on the popup.html will have the same effect. By implementing the search button so that it first directs the user to the Facebook search, the user will be directed to a friend's timeline rather than someone else who has the same or a similar name. The only issue is that if the user has two friends with the same name, as this extension will consistently go to one friend's timeline and never the other. In this case, the user must manually go to the second friend's timeline, and then click "Change Page" on the extension to move on to the next step (which is run by page2.html).

On page2.html (run by page2.js and after clicking "Search", "Enter", or "Change Page"), hovering over "Likes" will trigger a dropdown menu that has additional button. When a like button is clicked, page2.js sends a message to content.js which is the file that can interact directly with the website. Content.js will first select all the link tags and then go through each link with a for loop and search for the appropriate title attribute that corresponds to either posts, comments, or photos. Content.js will know which attribute to look for based on the subject of the message sent by page2.js. Whenever content.js finds a link with the correct title, it will like the link and move on. When all the links are liked, content.js will send a response back to page2.js via callback function (if requested by page2) which will change the page2.html and also display the "undo" button. "Undo" is a button on page 2 of the extension, and clicking it will send a message to content.js, unliking the latest likes made in the current session of the extension.