Appendix: B

Source code for each file:

Background.js:

```
'use strict';

let notifications = 0;

// run code on install/reload
chrome.runtime.onInstalled.addListener(function() {

	console.log(' background.js loaded');


	// test save urls

	chrome.storage.sync.set({"AmazonURLS": {"https://www.amazon.com/Amazon-Echo-
Dot-Portable-Bluetooth-Speaker-with-Alexa-
Black/dp/B01DFKC2SO/ref=zg_bs_electronics_home_3?
_encoding=UTF8&psc=1&refRID=B2NH4HN9QXK5K1D9KW8C":"4.43"}}, function(data){
		loadUrls();
	});

	chrome.browserAction.setBadgeText({text: ""});


});

// run code on extension startup
chrome.runtime.onStartup.addListener(function(){
	console.log('running on start up');
	loadUrls();
});




// Notes:
// the id priceblock_ourprice outputs the main price of a page.  However this seems to vary for
book listings which have multiple purchase options.  Maybe figure out how to test difference
or just ignore books?


// Scrapes the price of an amazon price product when given a url
function scrapePage(url){

	var xhttp = new XMLHttpRequest();
	xhttp.onreadystatechange = function() {
		if (this.readyState == 4 && this.status == 200) {

				// Parse using DOM Parser
				let parser = new DOMParser();
				let htmlDoc = parser.parseFromString(xhttp.responseText,
"text/html");

				// For book listings there are multiple prices.
				let price = ""
				try {
					price =
htmlDoc.getElementById('priceblock_ourprice').innerHTML;
				}
				catch(err) {
					try{
						price =
htmlDoc.getElementById('priceblock_dealprice').innerHTML;
					}
					catch(err){
						alert('Sorry I couldn\'t find the price of this product.');
					}
				}


				//remove any dollar signs ($) that may mess with parsing the string
to a float.
				let formatPrice = price.replace("$", "");

				console.log(formatPrice);

				// save new price.
				saveNew({[url]:formatPrice});

			}
		};
	xhttp.open("GET", url, true);
	xhttp.send();
}


// example amazon urls for testing purposes
function saveNew(test){
	// {"urlhere": "18.95"}
	// only one at a time for now

	chrome.storage.sync.get({"AmazonURLS": {}}, function(data){
		console.log(Object.keys(data.AmazonURLS));

		console.log(Object.keys(test)[0]);

		// save unique urls
		if(Object.keys(test)[0] in data.AmazonURLS){

			console.log("URL already stored.");
			for (let key in test){
				let tempPrice = test[key];
				console.log("Temp price is " + tempPrice);
				let tempUrl = key;
				console.log("Temp url is " + tempUrl);
				comparePrice(tempUrl, tempPrice);
			}

		} else {

			for(var key in test){
				let value = test[key];

				let newData = data.AmazonURLS;

				// add the new value to data
				newData[key] = value;
				chrome.storage.sync.set({"AmazonURLS": newData});

			}
		}

	});
}

// compares old amazon price with the new amazon price and then saves new price if it is
lower.
function comparePrice(url, newPrice){

	chrome.storage.sync.get({"AmazonURLS": {}}, function(data){

		if (url in data.AmazonURLS){
			// make saved price a float
			let oldPrice = data.AmazonURLS[url];
			console.log("The old price is " + oldPrice);
			if(newPrice < oldPrice){

				//update new price
				let newData = data.AmazonURLS;
				newData[url] = newPrice;
```

```
                        //notification of new deal
                        //chrome.browserAction.setBadgeText({text: "!"});
                        //alert('price change friendo');

                        // fun stuff

                        let notify = confirm("There has been a new deal detected!  Price
drop from " + oldPrice + " to " + newPrice + ".  Do you want to view product?");
                        if (notify == true){
                              chrome.tabs.create({url:url});
                        }

                        chrome.storage.sync.set({ "AmazonURLS" : newData });


                        // check that save worked
                        chrome.storage.sync.get({"AmazonURLS": {}}, function(data){
                              console.log(data.AmazonURLS);
                        });



                  } else {
                        console.log("no price change for " + url);
                        //TESTING
                        //loadUrls();
                  }


            } else {
                  console.log("There is no saved history of tracking " + url);
            }
      });

}






// Load saved urls

function loadUrls(){
      chrome.storage.sync.get({"AmazonURLS": {}}, function(data){
            let keyLength = Object.keys(data.AmazonURLS).length;
            console.log(data.AmazonURLS);
```

```
		// loop through array
		for(let key in data.AmazonURLS){
			// check price for each url
			scrapePage(key);
		}

	});
}




// GRAB TAB URL
chrome.extension.onConnect.addListener(function(port){
	console.log("Message recieved.  Sending response.");

	port.onMessage.addListener(function(msg){
		if (msg == "TABURL"){


			// initialize variable to hold the url in this context
			let tabURL = "";

			chrome.tabs.query({lastFocusedWindow: true, active: true}, function(tabs)
{
				// test if url can be retrieved
				if(tabs[0].url != undefined){
					console.log(tabs[0].url);
					tabURL = tabs[0].url;

					// do stuff with this new data
					if (tabURL.startsWith("https://www.amazon.com/")){
						console.log("Amazon Link Recieved.");
						scrapePage(tabURL);
					} else {
						console.log("NON-AMAZON LINK.");
						// return err msg;
						alert("Sorry this is a non-amazon page so I can't track
this product.")
						port.postMessage("NONAMAZON");
					}
				}
			});


		}
	});
```

```
});
```

manifest.json:

```
{
  "name": "Amazon Price Tracker",
  "version": "1.0",
  "description": "Tracks Amazon Product Prices",
  "manifest_version": 2,
  "background": {
        "persistent": false,
        "scripts":["background.js"]
  },
  "browser_action": {
        "default_popup": "popup.html",
        "default_title": "Track Amazon Products!"
  },
  "icons": {
        "32": "images/logoA32.png",
        "48": "images/logoA48.png",
        "128": "images/logoA128.png"
  },
  "permissions": [
        "storage",
        "tabs",
        "<all_urls>"
  ]
}
```

popup.css

```
/*
==============================
= Amazon Tracker Style Sheet =
==============================
====== By Milan Donhowe ======
==============================
*/



/*
===================
= Page-Wide Styles =
===================
*/

* {
   font-family: 'Lato', sans-serif;
```

```
    font-size:130%;
    text-align:center;
    font-weight: bold;
    color:#000;
    /*background-color: hsl(61, 100%, 50%);background-color: rgba(244, 188, 66, 0.8);*/
    background-color:#f7f7f7;
    padding:0px;
    margin:0px;
    /*was 180px*/
    width:180px;
}
/*
=================
= Button Styles =
=================
*/
.btn{
    border: 0px;
    border-top:0px black dashed;
    border-bottom:0px black dashed;
    margin-top:10px;
    width:60%;
    background-color:rgba(244, 188, 66, 0.8);
    border-radius:6px;
    text-align:center;
    color:#000;
    letter-spacing:0.02em;
}
.btn:hover{
    background: #e8c476;
    color:#fff;
}

/*
=====================
= Header Tag Styles =
=====================
*/

h1 {
    color:#000000;
    background-color:#FFFFFF;
    font-display: bold;
    text-decoration: none;
}

h2{
    border-top:0px black dashed;
    padding:2px;
    margin-top:20px;
    font-size:10px;
    background-color:#FFFFFF;
}

h6 {
    font-size:9px;
    background-color:#FFFFFF;
}



/*
================
= Table Styles =
================
*/


/*NOTE TABLE styles are mostly controlled by inline stylesheet for products.html*/
td{
    font-size:9px;
    border:1px black solid;
}

a {
    font-size:inherit;
}
```

popup.html

```
<!DOCTYPE HTML>

<html>

<head>
    <title>Amazon Price Tracker</title>
    <link type="text/css" rel="stylesheet" href="popup.css">
    <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">
</head>

<body>

    <h1>Amazon Price Tracker</h1>

            <a href="products.html"><button class="btn" id="load">Check Saved
Products</button></a>
```

```
    <button class="btn" id="save">Save This Product</button>


    <script src="popup.js"></script>

        <h2>Extension By <a href="https://github.com/MilanDonhowe" target="_blank">Milan
Donhowe</a> 2018</h2>

</body>

</html>
```

popup.js

```
/*
================================
=====- Popup.js script =======
================================
====== By Milan Donhowe ======
================================
*/


'use strict';

window.onload = function(){

    // attach functions to buttons on HTML
    var loadListen = document.getElementById("load").addEventListener('click', loadUrls);
    var saveListen = document.getElementById("save").addEventListener('click', savePage);


}


let loadUrls = () => {
    // load saved urls and iterate through them
    console.log("loadUrls called");
}

let savePage = () => {
    // get the current page, check if amazon product and if price is extractable.
    console.log("savePage called");

    let tabURL = "loading...";

    // ask background.js to do the heavily lifting
    let port = chrome.extension.connect({
        name: "Get Current Tab URL"
    });



    port.postMessage("TABURL");
    port.onMessage.addListener(function(msg){
        console.log("Message recieved:" + msg);
        tabURL = msg;
        // switch case for future expandability
        switch(msg){
            case "NONAMAZON":
                console.log("ERROR: Page not amazon.");
                break;
        }

    });


}
```
products.html
```
<!DOCTYPE HTML>

<html>

<head>

    <title>Amazon Price Tracker</title>

    <link type="text/css" rel="stylesheet" href="popup.css">

    <link href="https://fonts.googleapis.com/css?family=Lato" rel="stylesheet">

    <style>

      /*
      =====================================
      = Listening page specific styling =
      =====================================
      */

      html {
          width: 200%;
      }

      td {
       width: 150%;
       font-size: 20px;
      }

      .del {
         color:red;
      }

      .del:hover{
         color:blue;
         background-color:red;
      }

    </style>
</head>

<body>

    <h1>Amazon Price Tracker</h1>
    <h3>Products Listing</h3>
    <div id="productList">
       <table id="table"></table>
    </div>

    <a href="popup.html"><button class="btn">Back</button></a>
    <h2>Extension By Milan Donhowe 2018</h2>

    <script src="products.js"></script>
</body>

</html>
```

products.js

```
/*
================================
===== Product.js script ======
================================
====== By Milan Donhowe ======
================================
*/



// Loads saved urls
window.onload = function(){

    //var table = Document.createElement('p');
    chrome.storage.sync.get({"AmazonURLS": []}, function(data){

        for ( let i in Object.keys(data.AmazonURLS) ){

            let url = Object.keys(data.AmazonURLS)[i];
            let price = data.AmazonURLS[Object.keys(data.AmazonURLS)[i]];


            // regex recipe:
            var findName = /(?<=www.amazon.com\/)\w*/

            let name = url.match(findName);
                    let newHTML = "<tbody> " + "<tr><td><a href=" + url + " target='_blank'>" +
String(name)      +      "</a></td>"      +      "<td>"      +      String(price)      +      "</td><td
class='del'>X</td></tr></tbody>";
            let listing = document.getElementById('table').innerHTML += newHTML;

        }


        //add event listeners to delete them!


        let listRows = document.getElementsByClassName('del');
        let entry = 0

        while (entry < listRows.length){
            // test if entry is numerical
            if (typeof(entry) == typeof(3)){
                arg = listRows[entry];
                listRows[entry].addEventListener('click', function(arg){

                    // get the url
                    let tURL = arg.srcElement.parentNode.firstChild.lastChild.href;
                    // remove from storage with url name


                    chrome.storage.sync.get({"AmazonURLS": {}}, function(data){


                        //console.log(Object.keys(data.AmazonURLS));

                        for (var key in Object.keys(data.AmazonURLS)){
                            if (tURL == Object.keys(data.AmazonURLS)[key]){
                                delete data.AmazonURLS[tURL];
                            }
                        }
                        let newData = data.AmazonURLS;
                        chrome.storage.sync.set({"AmazonURLS":newData});

                    });


                    // delete HTML
                    arg.srcElement.parentElement.innerHTML = '';
                });
            }
            entry += 1;
        }



    });

}
```