<https://www.youtube.com/watch?v=uV4L-wcnK3Y&t=27s>

Can you write give me a code for a google chrome extension that can perform the following task: extract top new stories from major new outlets in the United States?

// The following code is written in JavaScript and can be used to create a Google Chrome extension to extract top news stories from major news outlets in the United States.

// Create a function to get the top stories

function getTopStories() {

// Create an array of URLs for major news outlets

const newsOutlets = [

"https://www.https://apnews.com/",

"https://www.startribune.com",

"https://www.nytimes.com/",

"https://www.foxnews.com/",

https://www.wsj.com/,

];

// Loop through the news outlets

for (let i = 0; i < newsOutlets.length; i++) {

// Create a new request

let xhr = new XMLHttpRequest();

// Request the news outlet's web page

xhr.open("GET", newsOutlets[i]);

// Send the request

xhr.send();

// Wait for the response

xhr.onreadystatechange = () => {

// Verify the response is successful

if (xhr.readyState == XMLHttpRequest.D

ONE && xhr.status == 200) {

// Get the response text

const responseText = xhr.responseText;

// Parse the response text for the top stories

const topStories = parseTopStories(responseText);

// Log the top stories

console.log(topStories);

}

};

}

}

// Create a function to parse the top stories

function parseTopStories(responseText) {

// Create an array to hold the stories

const topStories = [];

// Split the response text into an array of elements

const elements = responseText.split("<");

// Loop through the elements

for (let i = 0; i < elements.length; i++) {

// Get the element

const element = elements[i];

// Check if the element contains the string "story-heading"

if (element.includes("story-heading")) {

// Split the element into an array of words

const words = element.split(" ");

// Loop through the words

for (let j = 0; j < words.length; j++) {

// Get the word

const word = words[j];

// Check if the word contains the string "story-heading"

if (word.includes("story-heading")) {

// Get the story title

const storyTitle = words[j + 1];

// Add the story title to the array

topStories.push(storyTitle);

}

}

}

}

// Return the array of top stories

return topStories;

}

Here's a step-by-step guide to create a Chrome extension:

Choose a unique name and create a folder for your extension.

Create a file called manifest.json in the folder and add the following code:

json

Copy code

{

"manifest\_version": 2,

"name": "My Extension",

"version": "1.0",

"description": "A description of my extension",

"browser\_action": {

"default\_icon": "icon.png",

"default\_popup": "popup.html"

},

"permissions": [

"activeTab"

]

}

Create a file called popup.html in the folder and add the HTML code for the popup window of your extension.

Create an icon for your extension (a 128x128 PNG image) and save it as icon.png in the folder.

Go to chrome://extensions in Google Chrome and turn on "Developer mode" in the top right corner.

Click on the "Load unpacked" button and select the folder for your extension.

Your extension should now be loaded and visible in the browser. You can click on the extension icon to open the popup window.

You can add functionality to your extension by writing JavaScript code and linking it to the popup window using <script> tags in popup.html.

That's it! This is just a basic example, and you can add more features and functionality to your extension by exploring the Chrome extension API.

Here's an example of the structure of a manifest.json file:

json

Copy code

{

"manifest\_version": 2,

"name": "My Extension",

"version": "1.0",

"description": "A description of my extension",

"browser\_action": {

"default\_icon": "icon.png",

"default\_popup": "popup.html"

},

"permissions": [

"activeTab"

]

}

The manifest\_version field specifies the version of the manifest file format. The current version is 2.

The name field is the name of your extension.

The version field is the version number of your extension.

The description field is a brief description of your extension.

The browser\_action field specifies the default icon and popup window of your extension.

The permissions field specifies the permissions that your extension needs in order to run. In this example, the activeTab permission is needed to access the currently active tab in the browser.

Note that this is just a basic example, and there are many other properties and settings that you can specify in the manifest.json file to customize your extension. You can find more information about the Chrome extension manifest in the Chrome extension documentation.