Statement of Work (SOW)

Client Name: PeopleCaddie

Freelancer Name: AWS IQ Engineer

Project Title: LinkedIn Scraper Chrome Extension

Project Overview:

* The goal of this project is to develop a Chrome extension that can scrape a single profile on LinkedIn and save the data in a CSV file on AWS S3. The extension should be developed using a combination of Python and JavaScript.

Project Deliverables:

* A live demo of the Chrome browser plug-in with our DevOps engineer
* All configuration files
* Brief architectural notes (1-page)

Scope of Work:

* Develop a Chrome extension that can scrape a single profile on LinkedIn and save the data in a CSV file on AWS S3
* The extension should be developed using a combination of Python and JavaScript
* The extension should be able to scrape the name, experience, activity, education, and skills sections of a LinkedIn profile
* The extension should be able to save the scraped data in a CSV file on AWS S3 using the "boto3" module in Python
* The extension should be able to run on a paid enterprise LinkedIn user account
* The extension should be able to be loaded using the "Load unpacked" feature in the Chrome browser
* The project should be completed within 1 week of start time
* The freelancer will require access to the paid enterprise LinkedIn user account to test the extension

Budget:

* The budget for this project is a fixed price of $1000.00 - $1200.00 (flexible).

Timeline:

* The recommended completion date for this project is 3/10/2022.

Terms and Conditions:

* The freelancer will provide regular updates on the progress of the project
* The freelancer will maintain the confidentiality of all client information
* The client will provide the freelancer with access to the paid enterprise LinkedIn user account to test the extension
* The client will provide all necessary information and resources to the freelancer to complete the project
* The client will pay the freelancer the full amount of the project fee upon completion of the project

Acceptance:

* By accepting this SOW, the client agrees to the terms and conditions outlined in this document. The freelancer agrees to complete the project within the specified timeline and budget.

Sudo Code:

* Please see sample code for your workflows. This is not production code but just a framework for what we expect to be designed. Again, this just offers a framework and should not be used to complete the project.

To develop a Chrome extension to scrape a single profile on LinkedIn, you might need to use a combination of Python and JavaScript. Here are a few steps you might follow:

1. Create a new directory for the project and create a new file named manifest.json. This file is used to define the structure and permissions of the extension. Here's an example of what the file might look like:

| {  "manifest\_version": 2,  "name": "LinkedIn Scraper",  "version": "1.0",  "description": "Scrapes data from LinkedIn",  "permissions": [  "tabs",  "activeTab",  "http://\*/\*",  "https://\*/\*",  "storage",  "https://\*.linkedin.com/\*"  ],  "background": {  "scripts": ["background.js"]  },  "content\_scripts": [{  "matches": ["\*://\*.linkedin.com/in/\*"],  "js": ["content.js"]  }] } |
| --- |

1. In the same directory, create a new file named background.js. This file is used to define the background script for your extension. Here's an example of what the file might look like:

| chrome.browserAction.onClicked.addListener(function(tab) {  chrome.tabs.executeScript({  file: 'content.js'  }); }); |
| --- |

1. Create a new file named content.js, this could be written in python or any language you choose. This file is used to define the content script for your extension. Here's an example of what the file might look like:

| chrome.runtime.onMessage.addListener(function(request, sender, sendResponse) {  if (request.action == "scrapeProfile") {  var name = document.querySelector("li.inline.t-24.t-black.t-normal.break-words").innerText;  var experience = document.querySelector("section#experience-section").outerHTML;  var activity = document.querySelector("section#activity-section").outerHTML;  var education = document.querySelector("section#education-section").outerHTML;  var skills = document.querySelector("section#skills-section").outerHTML;    chrome.storage.sync.set({  "name": name,  "experience": experience,  "activity": activity,  "education": education,  "skills": skills  }, function() {  sendResponse({result: "success"});  });  } });  chrome.runtime.sendMessage({action: "scrapeProfile"}, function(response) {  console.log(response.result); }); |
| --- |

* 1. The sudo code above would scrape the name, experience, activity, education, and skills sections of a LinkedIn profile and save them to Chrome storage. The data will be sent back to the background script as a response.

1. In your Python script, you could use the "boto3" module to upload the data to AWS S3. Here's an example of what the code might look like:

| import boto3  s3 = boto3.resource('s3')  def uploadToS3(name, data):  bucket = s3.Bucket('my-bucket')  key = name + '.csv'  bucket.put\_object(Key=key, Body=data)  name = 'John Doe' data = 'Experience,Activity,Education,Skills\n' # add the scraped data  uploadToS3(name, data) |
| --- |

This code will upload the scraped data to AWS S3 in CSV format.

1. Finally, to run the extension, navigate to chrome://extensions in your Chrome browser and click "Load unpacked". Select the directory where your extension files are located, and your extension should now be loaded. To use the extension, a LinkedIn user would open a LinkedIn profile and click the extension button. The extension would scrape the profile.