**Startup Hunter**

**Author: Hamdan Sulaiman**

**Created on: 26th April 2021**

**Last Updated: 26th April 2021**

**Changes:**

* Goal 1.a) complete

**Overview:**

As a startup investor, it is very helpful to have all the up-to-date information about both a specific startup, as well as all new startups that are coming out. We will build a program that allows scraping the information about startups from specific websites.

Overtime we intend to a) Discover new startups in bulk and b) Dig into a specific startup and find out everything new there is to know about it.

**Goals:**

1. We will start this program by first displaying all the information on the console:
   1. Build a function that takes a url as parameter and captures a screenshot of a website and saves it in the current director with the filename being [website name]-[date].jpg
   2. Build a function that takes a url as parameter, and searches crunchbase.com for that website and finds out and prints everything we can know from CrunchBase about that company.
2. Over time this program will feed the information directly into a tool like Google Sheets or Air Table, or some other online platform.

**Solutions:**

* **Proposed Solution:**
  + **Default Browser Details:** By importing HKEY\_CURRENT\_USER, OpenKey, QueryValueEx from winreg we can get the name of the user’s default browser, based on the result, we can choose which webdriver to install.
  + **Using Selenium, alongside webdriver\_manager:** The selenium.webdrivermodule provides all the WebDriver implementations. Currently supported WebDriver implementations are Firefox, Chrome, IE and Remote. The webdriver\_manager checks if the files exist for installation, if not it downloads them
    - **Step 1:** The following line of code can be used to install the user’s default browser’s webdriver webdriver.Chrome/Firefox(ChromeDeviceManager/GeckoDriverManager.install. This also checks if the webdriver exists, if not it will install it.
    - **Step 2:** Minimize the empty browser window that opens.
  + **Prompting User:** We can prompt the user to enter the name of the website and then concatenate https//:www. and .com/
  + **Getting Screenshot:** 
    - **Step 1:** We can use the .get(url) method on the installed driver, to open the web page in the empty browser. We can use time.sleep(10) to allow the webpage to fully load before we take a screenshot.
    - **Step 2:** We can use try and except block to catch the WebDriverException in case the url is invalid.
    - **Step 3:** We can save the screenshot using driver.save\_screenshot(filename) and then close the browser window using driver.close().

**Results:**

* **Goal 1.a)** 
  + **Text

    Description automatically generatedInstalling WebDriverManager** according to user’s default browser, in this case the default browser is Firefox.
  + **Text

    Description automatically generatedPrompting User,** whilst only asking for the name of the website, in this case the name is tesla.

**Text

Description automatically generated**

* **A picture containing graphical user interface

  Description automatically generatedSaving Screenshot,** the screenshot is in the same directory as the code, it is saved as a png, though it can be saved as jpg as well but that gives a warning.

**Error Handling:** Entering the wrong web page name such as tsla instead of tesla raises a WebDriverException which is dealt with accordingly.

Graphical user interface, text

Description automatically generated

**GitHub Code:**

* https://github.com/msulaim/API-and-REST/blob/main/StartUp%20Hunter/startup-hunter.py