**Step-by-Step Guide to Checking 10L Mobile Number Registration on Amazon**

**1. Prepare Your Environment:**

* + First things first, make sure you have Python installed on your computer. You’ll also need a few Python libraries like selenium and webdriver\_manager to interact with web pages and manage the WebDriver

1. **Understand the Target Web Page:**
   * You’ll be working with Amazon India’s registration page. The task involves filling out the form on this page with a name, mobile number, and password, and then clicking the 'Continue' button to see if Amazon flags the mobile number as already registered.
2. **Set Up a Proxy Server for IP Rotation:**
   * To prevent getting blocked by Amazon for sending too many requests from the same IP, you'll need to rotate IP addresses. This can be achieved by using a proxy server that can handle IP rotation. This step is crucial to avoid rate-limiting or getting flagged by Amazon's anti-bot systems.
3. **Set Up Selenium WebDriver with Proxy:**
   * Now, you’ll need to configure Selenium WebDriver to use the proxy server. This involves setting up Chrome options to add the proxy server. Using a proxy helps rotate your IP address with each request, which mimics more natural user behavior.
4. **Create a Function to Check Mobile Number Registration:**
   * Write a function check\_amazon\_registration that:
     + Initializes the WebDriver with the proxy settings.
     + Navigates to Amazon’s registration page.
     + Fills in the registration form using a dummy name, the mobile number you want to check, and a dummy password.
     + Clicks the 'Continue' button to submit the form.
     + Waits for the response and checks if there’s an alert message indicating the number is already registered.
5. **Read Mobile Numbers from a CSV File:**
   * Since you have a large list of mobile numbers (10 lakhs or 1 million), you’ll want to read these from a CSV file to automate the process. Use Python’s csv module to handle reading the mobile numbers.
6. **Save the Results to a CSV File:**
   * After checking each number, you’ll want to save the results in a CSV file. The results will show each mobile number and whether it is already registered or not. This will help keep track of which numbers are already associated with an Amazon account.
7. **Automate the Process for Multiple Numbers:**
   * Write another function, check\_multiple\_numbers, that goes through each number from the list, calls check\_amazon\_registration for each, and stores the results. Make sure to add random delays between requests to imitate human-like behavior and further reduce the risk of getting blocked.
8. **Test Your Code with a Small Batch:**
   * Start with a small batch of mobile numbers to see if your code works as expected. Make sure the WebDriver interacts with the web page correctly, the proxies are working, and the results are accurately logged.
9. **Handle Errors:**
   * There might be unexpected errors while interacting with the web page or due to network issues. Make sure to add proper error handling in your code so that it can skip errors and continue running for the remaining numbers.
10. **Run the Script on AWS EC2 Instance:**

* To run the script continuously and without interruptions, consider using an AWS EC2 Windows instance.

1. **Monitor the Script:**

* Keep an eye on the script to ensure it runs smoothly. AWS EC2 allows you to manage long-running tasks effectively.
* **Be Prepared for CAPTCHAs:** Amazon might display CAPTCHAs or other verification methods. You might need to adjust the script or handle these cases manually.

**Python Code**