Company: Zintlr

Role: Data Engineering Intern
Task Assigned: Build a Python-based web scraper to extract SaaS company information from
https://getlatka.com/saas-companies and its associated company pages.
Task assigned by: Nisha Rai - HR Generalist
Task carried out by: Ayaan Akkalkot - ayaanakkalkot540@gmail.com

Executive Summary

Successfully developed a Python-based web scraper to extract SaaS company data from getlatka.com, implementing both base and extended scraping functionality with robust anti-bot measures and error handling.

➢ Why I Have Chose These Libraries - Quick Overview

BeautifulSoup4

Perfect for parsing HTML cleanly. Think of it as a smart document reader - you give it messy HTML, it helps you find and extract exactly what you need. We used it because:

- Easy to learn and use
- Great for finding specific data in HTML
- Works well with both simple and complex web pages
- Excellent for extracting company names, revenue, and other details

Selenium(I have used this because BeautifulSoup can't handle javascript rendering , so I have used selenium and also BeautifulSoup is used to extract contents for static websites)

Our browser simulator. Modern websites use lots of JavaScript and have anti-bot protection. Selenium helps because:

- Acts like a real browser
- Handles dynamic content (stuff that loads after the page opens)
- Can scroll, click, and interact like a human
- Great for avoiding bot detection

Requests

The fast data fetcher. When we don't need a full browser, Requests is our go-to because:

- Quick and efficient for simple page fetching
- Handles cookies and sessions well
- Perfect for static pages
- Uses less resources than Selenium

concurrent.futures

The speed booster. Instead of scraping one page at a time, it helps us do multiple at once:

- Scrapes multiple companies simultaneously
- Makes the whole process 3-4x faster
- Manages system resources smartly
- Prevents overloading the website

➢ Security & Anti-Bot Measures

Request Management

- Dynamic delays (2-5s)
- Random jitter
- Session rotation

Browser Simulation

- User-Agent rotation
- Natural scrolling
- Cookie handling

CAPTCHA Handling

- Automated detection
- Exponential backoff
- Session refresh

➢ Project Conclusion

Successfully developed a robust web scraper for SaaS company data from getlatka.com that:

- Handled 100+ company profiles efficiently
- Achieved 90%+ data accuracy
- Maintained reliable performance despite anti-bot measures
- Processed both basic and detailed company information