3. Challenges Faced & Resolutions

Challenge: Finding deep URLs hidden in JavaScript

Impact: Many job pages were loaded dynamically via JavaScript and were missed by basic crawlers.
Resolution: Implemented regex-based scanning of JavaScript code blocks to detect keywords like *job*, *career*, *company*, and *location*. Subsequently, resolved relative paths to complete URLs to discover hidden pages.

Challenge: Avoiding duplicate or irrelevant links

Impact: Duplicate or irrelevant links increased crawl time and wasted resources.
Resolution: Utilized set() data structures to ensure unique URLs and incorporated strict filtering rules in the is_valid_url function to exclude non-HTML files, spammy URLs, and excessively long URLs.

Challenge: Performance issues on large websites

Impact: Crawling large sites led to slow execution and high memory consumption.
Resolution: Adopted deque for efficient queue operations, implemented crawling delay to control request rates, and set maximum URL limits to prevent infinite crawling loops.

Challenge: Redirect handling

Impact: Some URLs redirected to final destination pages that were not captured in the initial crawl list.
Resolution: Enabled automatic redirect handling in HTTP requests and ensured redirected URLs were re-queued for crawling if they passed validation.

Challenge: Risk of server overload and IP blocking

Impact: Excessive request rates risked server overload, leading to IP bans.
Resolution: Introduced delays between requests and limited the number of URLs generated and tested from dynamic URL patterns to reduce server load.