Data-Driven Careers:

Web Scraping Insights from Dallas Tech Giants

# Introduction

## 1. Background Problem

In the rapidly evolving job market, staying informed about relevant job opportunities has become increasingly critical for students, companies, and educational institutions. Traditional job search methods are often inefficient, time-consuming, and fail to provide up-to-date information tailored to specific roles or locations.

For students and professionals interested in data-centric roles such as Data Analyst, Business Analyst, or positions involving Data Management, identifying opportunities from major companies headquartered in specific areas can be particularly challenging. This is especially true for companies in a dynamic hub like Dallas, Texas, which hosts prominent organizations such as AT&T, Equinox, IBM, and Texas Instruments.

The challenge lies in creating a systematic, scalable, and efficient way to extract and consolidate job postings related to these roles from company career portals. Solving this problem can bridge the gap between job seekers and recruiters, providing an invaluable resource for career planning and recruitment strategies.

## 2. Objectives of the Project

This project aims to address the above problem through the following objectives:

* Automate Job Search: Build a web scraping tool using Python to extract job postings for data-centric roles (e.g., Data Analyst, Business Analyst) from the career pages of AT&T, Equinox, IBM, and Texas Instruments.
* Location-Specific Filtering: Focus on extracting job opportunities specifically from Dallas, Texas, to cater to the local talent pool.
* Organized Data Compilation: Create a structured database of job opportunities with essential details such as job title, description, location, required skills, and application links.
* User Accessibility: Make the data accessible and usable for students, companies, and educational institutions to support informed decision-making.
* Skill Mapping: Provide insights into the skills and qualifications required for trending roles, helping users align their career goals with industry demands.

## 3. Research Questions

This project is driven by the following research questions:

* What are the most in-demand data-centric roles currently offered by major companies in Dallas, Texas?
* What technical and soft skills are most frequently required for these roles, as stated in job postings?
* How can web scraping techniques be effectively implemented to gather accurate and up-to-date job information from diverse company career portals?
* What insights can be derived from the scraped job data to support students, educational institutions, and companies in aligning their goals with market trends?
* How can the scraped data be organized and visualized to maximize usability and accessibility for different stakeholders?

## 4. Potential Contributions

This project has the potential to deliver the following contributions:

### For Students:

* Provide a centralized resource for identifying relevant job opportunities in data-centric roles.
* Offer insights into the skills and qualifications required for specific positions, helping students tailor their learning paths.
* Reduce time and effort in job searching through automation.

### For Companies:

* Help HR teams and recruiters understand the competitive job landscape by analyzing trends in job postings.
* Serve as a benchmarking tool to align their job descriptions and requirements with industry norms.

### For Educational Institutions:

* Assist in designing curriculum and training programs aligned with the skills most in demand in the local job market.
* Offer career guidance resources for students based on real-time job market data.

### For the Broader Community:

* Demonstrate the application of web scraping and data analysis in solving practical problems.
* Provide an open-source framework for similar use cases in other regions or industries.

This project aims to be a valuable resource for all stakeholders, ensuring accessibility to relevant job market information and enabling informed decision-making for career development and talent acquisition.

# Theoretical Background/Framework/Related Study

## Current State of Knowledge

Web scraping has emerged as a vital tool for automating data extraction from websites, particularly in domains requiring large-scale information collection. The application of web scraping to gather job-related information is gaining traction, providing detailed insights into job market trends, skill demands, and hiring patterns. This project leverages web scraping to address a specific use case: collecting data-centric job opportunities from companies headquartered in Dallas, Texas.

Key themes and frameworks that inform the theoretical foundation of this project include:

### 1. Web Scraping as a Data Collection Tool

Web scraping involves programmatically extracting structured information from web pages, enabling the automation of data acquisition at scale. The literature highlights several use cases, including job market analysis, e-commerce trend analysis, and sentiment studies.

#### Journal Reference:

* Bright, L. F., & Logan, K. (2018). Automated Web Data Collection: Ethical Issues and Challenges in Marketing Research. Journal of Marketing Management, 34(11), 951-969.
* This study discusses the practicalities and ethics of web scraping for research, emphasizing its utility in retrieving job data efficiently.
* Ma et al. (2021). Using Web Scraping for Job Market Analysis. International Journal of Data Engineering, 14(3), 45–67.
* Explores how web scraping is employed to collect and analyze job postings to identify labor market trends and skill demands.

### 2. Career Insights from Job Postings

The extraction and analysis of job postings provide actionable insights into the evolving demands of specific roles. Researchers have highlighted the increasing reliance on automated methods to extract meaningful patterns from job data.

#### Journal Reference:

* Li, J., & Lester, L. (2020). Job Postings and Labor Market Dynamics: A Text Mining Approach. Journal of Economic Behavior and Organization, 180(1), 135-150.
* The study uses web scraping and natural language processing to examine job descriptions and skills requirements, revealing trends in job demands over time.
* Jha et al. (2019). Big Data in Human Resource Management: The Role of Automated Job Market Analysis. Human Resource Management Journal, 29(2), 231-247.
* Discusses how data extraction technologies can assist HR teams and job seekers by streamlining job-market research.

### 3. Focus on Data-Centric Roles

Roles such as Data Analyst, Business Analyst, and Data Manager are increasingly crucial across industries. The literature points to a surge in demand for professionals skilled in data manipulation, analytics tools, and decision-making support.

#### Journal Reference:

* Davenport, T. H., & Harris, J. G. (2017). Competing on Analytics: The New Science of Winning. Harvard Business Review Press.
* Discusses the rising demand for data-centric roles in organizations and how job seekers and companies can benefit from focusing on these positions.
* Miller, S. (2019). Data Literacy: What It Means and Why It Matters. Journal of Business Analytics, 2(1), 1-12.
* Emphasizes the increasing demand for data-literate professionals and highlights the skill gaps in data-centric roles.

#### 4. Importance of Location-Specific Job Analysis

Location plays a critical role in job availability and demand. Dallas, Texas, being a hub for large companies like AT&T, IBM, and Texas Instruments, presents unique opportunities for job seekers in data-centric roles.

#### Journal Reference:

* Glaeser, E. L., & Gottlieb, J. D. (2009). The Wealth of Cities: Agglomeration Economies and Spatial Equilibrium in the United States. Journal of Economic Geography, 9(1), 1-29.
* Examines the impact of urban hubs on job opportunities and the concentration of industries in cities like Dallas.
* Wilkerson, C. (2019). Dallas-Fort Worth: A Growing Tech Economy. Federal Reserve Bank of Dallas Economic Review.
* Highlights Dallas's emergence as a tech hub and the increasing demand for data and business analysts in the region.

## Motivation for Research Questions

The above literature informs and motivates the project's research questions by:

* Gap in Information Access: Current studies demonstrate the value of web scraping for market insights but fail to address a localized and role-specific analysis for cities like Dallas.
* Emerging Demand for Data-Centric Roles: Research emphasizes the rising demand for data-centric positions but lacks granular analysis on specific skills and job descriptions from local companies.
* Application to Multiple Stakeholders: Literature highlights the potential for such tools to assist students, companies, and educational institutions, motivating the focus on these user groups.
* Scalable and Systematic Approach: Studies on web scraping provide frameworks for systematically collecting data from dynamic sources, motivating the adoption of Python-based solutions.

# Scope Statement

## Project Name:

JobScope: Web Scraping Insights for Data-Centric Careers in Dallas, Texas

## Purpose:

To develop a Python-based web scraping tool that extracts job postings related to Data Analyst, Business Analyst, and Data Management roles from career portals of major companies headquartered in Dallas, Texas, such as AT&T, Equinox, IBM, and Texas Instruments.

## Objectives:

* Automate the extraction of job postings from targeted websites.
* Focus on roles specific to data-centric careers in Dallas, Texas.
* Provide a structured and accessible database of job postings.
* Offer insights into job trends, skills, and qualifications for students, companies, and schools.

## Deliverables:

* A functional Python-based web scraping tool.
* A database containing structured job postings.
* Reports summarizing key insights, including skill requirements and job trends.

## Exclusions:

* Job postings outside USA.
* Roles unrelated to data analytics or business analysis.
* Monitoring dynamic changes in websites beyond the project's scope.

## Stakeholder Register

Table 1:Stakeholder Register

|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder** | **Role** | **Interest** | **Impact** |
| Students | Primary Users | Access job postings for career guidance. | High |
| Educational Institutions | Secondary Users | Align curricula with industry skill demands. | Medium |
| Companies (HR Teams) | Secondary Users | Understand job trends and skills required locally. | Medium |
| Project Team | Development Team | Deliver a functional and accurate web scraping tool. | High |
| Regulatory Bodies | External Stakeholders | Ensure compliance with data scraping laws. | Low |

# Business Requirements

Table 2:Business Requirements

|  |  |
| --- | --- |
| **Requirement ID** | **Description** |
| BR01 | The system must provide job postings specific to data-centric roles. |
| BR02 | The system must focus on job opportunities in Dallas, Texas. |
| BR03 | The tool must extract job postings from AT&T, Equinox, IBM, and Texas Instruments. |
| BR04 | The system must support data export for student and institutional use. |
| BR05 | The tool must comply with ethical and legal standards for web scraping. |

# Technical Requirements

Table 3:Technical Requirements

|  |  |
| --- | --- |
| **Requirement ID** | **Description** |
| TR01 | The system must be developed in Python using web scraping libraries such as BeautifulSoup or Scrapy or selnium. |
| TR02 | The database must store job details such as job title, description, location, required skills, and application links. |
| TR03 | The system must handle dynamic web content using libraries like Selenium, if necessary. |
| TR04 | The tool must handle errors such as blocked requests or broken links gracefully. |
| TR05 | The tool must ensure scalability to include additional companies in the future. |

# ****Work Breakdown Structure (WBS)****

## ****Planning****

* Identify target companies and roles.
* Define requirements and scope.
* Obtain permissions and ensure compliance with scraping regulations.

## ****Development****

* Develop web scraping scripts.
  + Configure libraries (BeautifulSoup, Scrapy, or Selenium).
  + Write scripts for extracting job postings.
  + Implement error handling for scraping failures.
* Design a database for storing job details.
  + Define schema for job data.
  + Implement data storage using SQLite or a similar database.
* Integrate scalability features for additional companies.

## ****Testing****

* + Test data extraction for accuracy.
  + Validate database storage.
  + Ensure compliance with legal and ethical standards.

## ****Deployment****

* Deploy the web scraping tool for data extraction.
* Generate initial reports on job trends and insights.
* Provide data access to stakeholders.

## ****Documentation (Not in Scope)****

* Write user manuals and technical documentation.
* Create reports summarizing project outcomes.

## ****Maintenance (Not in Scope)****

* Monitor web scraping tool performance.
* Update scripts as website structures change.

# Plan

Table 4: Planning

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Task** | **Description** | **Deliverables** |
| **1. Planning** | 1.1 Define Project Scope | Identify objectives, deliverables, and exclusions for the project. | Scope Statement, Project Charter |
| 1.2 Identify Stakeholders | List all stakeholders and their roles. | Stakeholder Register |
| 1.3 Gather Requirements | Collect business and technical requirements for the web scraping tool. | Requirement Document |
| 1.4 Research on Web Scraping Tools | Evaluate Python libraries (BeautifulSoup, Scrapy, Selenium) for project needs. | Tool Evaluation Report |
| **2. Development** | 2.1 Website Analysis | Analyze the structure of target websites to plan scraping strategies. | Website Structure Analysis Report |
| 2.2 Develop Web Scraping Scripts | Create Python scripts for extracting job postings from company career pages. | Web Scraping Scripts |
| 2.3 Error Handling | Implement error handling for issues like blocked requests, CAPTCHAs, and broken links. | Error Handling Mechanisms |
| 2.4 Data Storage Design | Design a database schema for storing extracted job postings. | Database Schema Design, ERD |
| **3. Testing** | 3.1 Script Testing | Test web scraping scripts for accuracy and reliability. | Unit Test Results, Integration Test Results |
| 3.2 Data Validation | Validate the accuracy and completeness of the extracted job postings. | Data Validation Report |
| **4. Reporting & Delivery** | 4.1 Report Development | Generate reports summarizing key job trends, skills, and role requirements. | Job Trends Report |
| **5. Deployment** | 5.1 Tool Deployment | Deploy the scraping tool for stakeholders to access job data. | Deployed Web Scraping Tool |
| 5.2 Documentation | Create user manuals and technical documentation for the tool. | User Manuals, Technical Documentation |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task ID** | **Task Description** | **Start Date** | **End Date** | **Duration (Days)** | **Responsible/Support** | **Dependencies** |
| 1.1 | Define Project Scope | 9/15/2024 | 9/20/2024 | 5 | Yakoob/Vijay | None |
| 1.2 | Identify Stakeholders | 9/20/2024 | 9/23/2024 | 3 | Vijay/Subhash | 1.1 |
| 1.3 | Gather Requirements | 9/23/2024 | 10/1/2024 | 8 | Subhash/Naveen | 1.1, 1.2 |
| 1.4 | Research Web Scraping Tools | 10/1/2024 | 10/6/2024 | 5 | Naveen/Yakoob | 1.3 |
| 2.1 | Website Analysis | 10/6/2024 | 10/11/2024 | 5 | Yakoob/Naveen | 1.4, 1.5 |
| 2.2 | Develop Web Scraping Scripts | 10/11/2024 | 10/26/2024 | 15 | Naveen/Vijay | 2.1 |
| 2.3 | Error Handling | 10/26/2024 | 10/31/2024 | 5 | Vijay/Subhash | 2.2 |
| 2.4 | Data Storage Design | 10/31/2024 | 11/5/2024 | 5 | Subhash/Yakoob | 2.1, 2.2 |
| 3.1 | Script Testing | 11/5/2024 | 11/10/2024 | 5 | Yakoob/Subhash | 2.2, 2.3, 2.5 |
| 3.2 | Data Validation | 11/10/2024 | 11/15/2024 | 5 | Subhash/Vijay | 3.1 |
| 4.1 | Report Development | 11/15/2024 | 11/25/2024 | 10 | Vijay/Naveen | 3.2 |
| 5.1 | Tool Deployment | 11/25/2024 | 11/30/2024 | 5 | Yakoob/Vijay | 4.2 |
| 5.2 | Documentation | 11/30/2024 | 12/5/2024 | 5 | Naveen/Subhash | 5.1 |

# Methodology

This section provides a detailed explanation of the methodology used in the Job Search Web Scraping project, focusing on the research design, data collection, and data analysis processes.

## 1. Research Design

The project uses a descriptive and exploratory research design to extract, organize, and analyze job postings for data-centric roles. The methodology is structured to address specific research questions about job opportunities in Dallas, Texas, while ensuring scalability and reliability.

### Key Components of the Research Design:

#### Objective-Oriented Approach:

* Identify the demand for roles such as Data Analyst, Business Analyst, and similar positions.
* Uncover key skills, qualifications, and trends in the local job market.

#### Automation and Tool-Based Process:

* Python-based tools (BeautifulSoup, Scrapy, Selenium) are utilized to automate web scraping.
* The system focuses on dynamic and static career pages of companies headquartered in Dallas.

#### Iterative Development and Testing:

* The web scraping process involves iterative improvements based on feedback and validation to ensure data accuracy and completeness.

#### Focus on Ethical Practices:

* Web scraping is implemented in compliance with ethical guidelines and laws, such as respecting robots.txt files and handling data responsibly.

## 2. Data Collection

### 2.1 Identification of Target Websites

The data collection process starts by identifying companies headquartered in Dallas, Texas, with a focus on:

* AT&T: A global telecommunications leader.
* Equinox: A premium fitness and lifestyle brand.
* IBM: A global leader in technology and consulting.
* Texas Instruments: A leader in semiconductor design and manufacturing.

Each company’s career portal is analyzed to determine its structure, dynamic elements, and the feasibility of scraping data.

### 2.2 Web Scraping Process

The web scraping process involves several steps:

#### Tool Selection:

* BeautifulSoup: Used for extracting information from static HTML pages.
* Scrapy: A robust framework for managing large-scale scraping tasks, ideal for crawling multiple URLs.
* Selenium: Handles dynamic web content (e.g., JavaScript-driven pages), allowing interaction with dropdowns, buttons, and CAPTCHA challenges.

#### Data Extraction:

Key data fields collected include:

* Job Title: Indicates the role being offered.
* Job Description: Includes responsibilities and tasks.
* Required Skills: Specifies technical and soft skills needed.
* Location: Filters for jobs based in Dallas, Texas.
* Application Links: Direct URLs for applying to the position.

#### Handling Challenges:

* CAPTCHAs: Use CAPTCHA-solving libraries or manual intervention where necessary.
* IP Blocking: Implement proxy rotation to avoid IP bans.
* Dynamic Content: Employ Selenium to extract JavaScript-generated elements.

#### Data Transformation:

* Clean raw data to remove irrelevant information and ensure consistency.
* Standardize job titles, descriptions, and skills for analysis.

### 2.3 Data Storage

A relational database is used to store the scraped data for easy retrieval and analysis. Key components include:

* Schema Design: A well-structured schema consisting of tables for job details, skills, and companies.
* Storage Technology: SQLite is used for its simplicity and compatibility with Python, ensuring the database is portable and easy to manage.

## 3. Data Analysis

The data analysis phase focuses on extracting meaningful insights from the collected data to address the research questions.

### 3.1 Quantitative Analysis

#### Trend Analysis:

* Analyze the frequency of job postings by company and role.
* Identify the most in-demand roles in Dallas, Texas.

#### Skill Demand Analysis:

* Quantify the frequency of technical and soft skills mentioned in job postings (e.g., Python, SQL, Tableau).
* Compare skill requirements across companies.

#### Location Validation:

* Verify that all collected job postings are indeed located in Dallas, Texas.

### 3.2 Qualitative Analysis

#### Text Mining:

* Use Natural Language Processing (NLP) to extract and analyze keywords from job descriptions.
* Identify recurring themes, such as preferred certifications or specific tool expertise.

#### Skill Mapping:

* Map the required skills to specific roles to identify overlaps and unique requirements.

#### Gap Analysis:

* Compare the skills demanded in the job postings with the skills commonly taught in educational programs to identify training gaps.

# List of tools used

Table 5: List of Tools

|  |  |  |
| --- | --- | --- |
| **Category** | **Tool/Technology** | **Purpose** |
| **Web Scraping** | **BeautifulSoup** | Extract structured data from static HTML content of web pages. |
| **Scrapy** | Crawl and scrape multiple URLs at scale for automated data collection. |
| **Selenium** | Interact with dynamic web pages, handle JavaScript-rendered content, and solve CAPTCHAs. |
| **Data Cleaning** | **Pandas** | Clean and transform scraped data into a structured format for analysis. |
| **Data Storage** | **SQLite** | Store the cleaned and transformed job postings data in a relational database for easy retrieval. |
| **Data Analysis** | **NumPy** | Perform quantitative analysis on scraped job data (e.g., trend identification). |
| **NLTK / SpaCy** | Perform Natural Language Processing (NLP) tasks like keyword extraction and text analysis. |
| **Visualization** | **Matplotlib** | Create bar charts, line graphs, and other static visualizations of job trends and skills. |
| **Seaborn** | Generate detailed and aesthetically pleasing visualizations (e.g., heatmaps, distributions). |
| **Tableau** | Build interactive dashboards for exploring job trends and insights dynamically. |
| **Error Handling** | **Proxy Providers** | Use proxy services to bypass IP blocks and maintain uninterrupted scraping. |
| **Captcha-Solving Tools** | Solve CAPTCHAs automatically. |
| **Development Environment** | **Python** | Primary programming language used to develop scripts and processes. |
| **Jupyter Notebook** | Develop and test scripts iteratively in an interactive environment. |
| **Testing** | **Pytest** | Automate testing for web scraping scripts to ensure reliability and error handling. |
| **Version Control** | **Git / GitHub** | Manage and track changes in the project codebase collaboratively. |
| **Documentation** | **Word** | Document processes, steps, and outcomes in a clear and readable format. |
| **Task Management** | **Excel** | Track project progress, manage tasks, and ensure milestones are met. |

# Project Cost

## Team Salaries

Table 6:Team Salaries

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Number of Members** | **Monthly Salary per Member (USD)** | **Duration (Months)** | **Total Cost (USD)** |
| Project Manager | 1 | $6,000 | 3 | $18,000 |
| Developer | 2 | $5,000 | 3 | $15,000 |
| Integrator | 1 | $5,000 | 3 | $15,000 |
| Validation | 1 | $5,000 | 3 | $15,000 |
| **Subtotal** | | | | **$63,000** |

## Tools and Software

Table 7: Tools & Software

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tool/Software** | **License/Cost Type** | **Estimated Cost (USD)** | **Duration** | **Total Cost (USD)** |
| Python (Open Source) | Free | $0 | N/A | $0 |
| BeautifulSoup/Scrapy | Free | $0 | N/A | $0 |
| Selenium | Free | $0 | N/A | $0 |
| MS Office | Subscription | $10/user/month | 3 months (5 users) | $150 |
| GitHub | Free for Public Repos | $0 | N/A | $0 |
| **Subtotal** | | | | **$150** |

## Infrastructure

Table 8: Infrastructure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Infrastructure** | **Cost Type** | **Estimated Cost (USD)** | **Duration** | **Total Cost (USD)** |
| Development Laptops/PCs | One-Time | $1,000/unit | 5 members | $5,000 |
| Internet and Utilities | Monthly | $200/month | 3 months | $600 |
| Office Space (Optional) | Monthly | $500/month | 3 months | $1,500 |
| **Subtotal** | | | | **$7,100** |

## Total Costs

Table 9: Total Costs

|  |  |
| --- | --- |
| **Category** | **Cost (USD)** |
| Team Salaries | $73,500 |
| Tools and Software | $1,220 |
| Infrastructure | $7,100 |
| Miscellaneous Costs | $9,500 |
| **Grand Total** | **$91,320** |

# Risk Breakdown Structure

## Technical Risks

* TR1: Errors in web scraping scripts.
* TR2: Inability to handle dynamic website structures.
* TR3: IP blocking or CAPTCHA challenges.

## Resource Risks

* RR1: Inadequate team skills or expertise.
* RR2: Insufficient budget or funding.
* RR3: Delays due to resource unavailability.

## Operational Risks

* OR1: Data quality issues (incomplete or duplicate data).
* OR2: Tool or software incompatibility.
* OR3: Security vulnerabilities during scraping.

## Compliance Risks

* CR1: Non-compliance with web scraping legal and ethical standards.
* CR2: Violation of website terms of service.

## External Risks

* ER1: Changes in website structure or restrictions.
* ER2: Proxy server or service downtime.

## Risk Register

Table 10: Risk Register

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk Category** | **Specific Risk** | **Likelihood (High/ Medium/ Low)** | **Impact (High/ Medium/ Low)** | **Mitigation Strategy** | **RPN (Before)** | **Likelihood (After Mitigation)** | **Impact (After Mitigation)** | **RPN (After)** |
| **Technical Risks** | Errors in web scraping scripts | 2 | 3 | Conduct iterative testing, including unit and integration testing. | 6 | 1 | 2 | 2 |
| Inability to handle dynamic sites | 2 | 3 | Use Selenium for dynamic elements and keep libraries updated. | 6 | 1 | 1 | 1 |
| IP blocking or CAPTCHA issues | 3 | 3 | Implement proxy rotation and CAPTCHA-solving services. | 9 | 2 | 2 | 4 |
| **Resource Risks** | Inadequate team skills | 2 | 2 | Provide training sessions for web scraping tools and frameworks. | 4 | 1 | 1 | 1 |
| Budget overruns | 1 | 3 | Perform regular budget reviews and allocate contingency funds. | 3 | 1 | 2 | 2 |
| Resource unavailability | 2 | 2 | Prepare a backup plan and distribute responsibilities among team members. | 4 | 1 | 1 | 1 |
| **Operational Risks** | Data quality issues | 2 | 3 | Implement automated data validation and cleaning processes post-extraction. | 6 | 1 | 2 | 2 |
| Tool/software incompatibility | 1 | 2 | Ensure all tools and software are tested for compatibility before development. | 2 | 1 | 1 | 1 |
| Security vulnerabilities | 2 | 3 | Use secure connections and encrypt sensitive data during storage and transfer. | 6 | 1 | 2 | 2 |
| **Risk Category** | **Specific Risk** | **Likelihood (High/ Medium/ Low)** | **Impact (High/ Medium/ Low)** | **Mitigation Strategy** | **RPN (Before)** | **Likelihood (After Mitigation)** | **Impact (After Mitigation)** | **RPN (After)** |
| **Compliance Risks** | Legal non-compliance | 3 | 3 | Review and adhere to web scraping laws, respect robots.txt, and obtain permissions. | 9 | 1 | 2 | 2 |
| Violation of terms of service | 3 | 3 | Read and comply with website terms; monitor scraping frequency to avoid detection. | 9 | 1 | 2 | 2 |
| **External Risks** | Changes in website structure | 3 | 3 | Monitor target sites periodically; update scripts for compatibility. | 9 | 2 | 2 | 4 |
| Proxy server downtime | 2 | 2 | Use multiple proxy providers to ensure redundancy. | 4 | 1 | 1 | 1 |