# Job Scraper Tool Documentation

This Job Scraper Tool is designed to simplify the process of scraping and analyzing data from multiple job boards, providing essential information for job seekers and recruiters. This document will overview the tool's architecture, components, and user-interface.

## Overview

The Job Scraper Tool is a web application consisting of a user-interface (frontend) built with HTML, CSS, and JavaScript, and a backend server powered by Python. The application scrapes job listings from various job boards such as careergov, Indeed, and Naukri. Once the data is scraped, it is consolidated and displayed in a user-friendly interface, which allows users to view and analyze job trends such as popular categories, salary distributions, and more.

## Components

The components of the project include:

- \*\*Python Scraping\*\*: The backend server implemented using Python and the Quart library scrapes data from specified job boards using custom scraping functions tailored for each site.

- \*\*Model for Unsupervised Learning of Category Clusters\*\*: The Job Title Clustering feature groups job titles into clusters based on their textual similarity using unsupervised machine learning techniques – K-Means clustering, in particular.

- \*\*UI for Display\*\*: The frontend user interface built with HTML, CSS, and JavaScript displays the scraped and processed data in a user-friendly format in form of tables and charts.

- \*\*Response Code Error Handling\*\*: Error handling is implemented in the backend server to respond with appropriate HTTP status codes for successful or unsuccessful requests.

- \*\*Documentation\*\*: This document provides an overview and explanation of the tool's features and functions.

## User Interface

The web interface is organized into four main tabs:

1. \*\*Scrape Data\*\*: This tab offers buttons for scraping job listings from careergov, Indeed, and Naukri. Users may choose to scrape data from individual sites or all three.

2. \*\*Consolidate Data\*\*: This tab consolidates the scraped data from different job boards into a single dataset.

3. \*\*Display Data\*\*: The consolidated data is displayed in a table format to explore individual job listings in this tab.

4. \*\*Statistics and Trends\*\*: This tab presents insightful charts and graphs based on the consolidated job data, including job categories, average salaries, and experience levels. The data is further partitioned into clusters based on job title similarity.

## Python Scripts Overview

The project includes the following Python scripts:

1. \*\*app.py\*\*: Defines the server endpoints for the web application and handles frontend requests.

2. \*\*process\_data.py\*\*: Provides functions to process and consolidate the scraped data from different job boards.

3. \*\*job\_title\_clustering.py\*\*: Implements the K-Means clustering algorithm to group similar job titles into clusters.

4. \*\*scraping\_careergov.py\*\*: Scrapes and processes job listings from careergov.

5. \*\*scraping\_indeed.py\*\*: Scrapes and processes job listings from Indeed.

6. \*\*scraping\_naukri.py\*\*: Scrapes and processes job listings from Naukri.

## Usage

1. Install the required Python libraries for the project using pip install.

2. Run the web application server using `python app.py`.

3. Visit the web application by opening a web browser and navigating to `http://localhost:5000`.

4. Use the Scrape Data tab to scrape job listings from selected job boards.

5. Press the Consolidate Data button to consolidate all the scraped data into a single dataset.

6. Navigate to Display Data to explore the job listings in a table format.

7. Use the Statistics and Trends tab to gain insights into job categories, salaries, and other job-related trends.

8. Optional: Adjust the clustering algorithm through the job\_title\_clustering.py script to change the number of clusters or input custom parameters.