**WEB SCRAPING SYLLABUS**

**1. Introduction to Web Scraping**

* What is web scraping?
* Use cases and real-world applications
* Legal & ethical considerations (robots.txt, terms of service)

**2. Understanding HTML & CSS Basics**

* HTML structure (tags, attributes, elements)
* Common elements (div, span, table, a, img, etc.)
* CSS selectors for targeting elements
* Using browser Developer Tools (Inspect Elements)

**3. Regular Expressions (Regex) for Web Scraping**

* Introduction to regular expressions
* Basic regex syntax (., \*, +, ?, ^, $, \d, \w, etc.)
* Matching patterns in scraped data
* Extracting email addresses, phone numbers, prices, and dates
* Using regex with Python (re module)
* Applying regex in BeautifulSoup & Scrapy

**4. Setting Up the Web Scraping Environment**

* Installing required libraries (requests, BeautifulSoup, Scrapy, Selenium)
* Choosing the right tool for the job

**5. Fetching Web Pages with HTTP Requests**

* Using requests to make GET and POST requests
* Handling response codes (200, 404, 403, etc.)
* Parsing JSON responses for API data

**6. Parsing Web Pages with BeautifulSoup**

* Loading and parsing HTML
* Finding elements with find(), find\_all()
* Using CSS selectors and XPath in BeautifulSoup
* Extracting text and attributes (.text, ['href'])

**7. Handling Dynamic Websites with Selenium**

* Introduction to Selenium and WebDriver
* Automating browser interactions (clicks, scrolling)
* Handling JavaScript-rendered content
* Taking screenshots and extracting dynamic data

**8. Using Scrapy for Large-Scale Scraping**

* Scrapy project structure
* Writing a spider to scrape multiple pages
* Handling pagination and AJAX requests
* Exporting scraped data (CSV, JSON, databases)

**9. Handling Authentication & Sessions**

* Logging into websites using sessions (requests.Session())
* Managing cookies and headers
* Handling CAPTCHA and bot detection

**10. Dealing with Anti-Scraping Measures**

* Understanding robots.txt
* Using headers and user agents
* Implementing request delays and rate limiting
* Using proxies and rotating IPs (scrapy-rotating-proxies)

**11. Storing & Processing Scraped Data**

* Saving data in CSV, JSON, and databases (SQLite, MongoDB, MySQL)
* Cleaning and formatting scraped data (pandas, regex)

**12. Web Scraping Best Practices**

* Avoiding excessive requests
* Respecting website policies (robots.txt, request limits)
* Keeping scripts efficient and maintainable

**13. Real-World Web Scraping Projects**

* Scraping e-commerce product data
* Extracting news articles
* Monitoring stock prices
* Collecting social media data