# **Python Project -Web Scraping On Flipkart**

**Website Structure:** flipkart website structure typically involves HTML markup with CSS classes and sometimes dynamic content loaded viva java script. Identify the specific page (e.g product listing ,search result ) you intend scrap and analyze their structure using browser developer tools.

### Tools and Libraries

**BeautifulSoup**:

* 1. **Overview**: A library for parsing HTML and XML documents.
  2. **Strengths**: Simple to use, excellent for beginners, effective for smaller projects.
  3. **Weaknesses**: Slower for large-scale scraping, requires an additional library like requests to fetch HTML content.

**Scrapy**:

* 1. **Overview**: An open-source and collaborative web crawling framework for Python.
  2. **Strengths**: Built specifically for web scraping, highly efficient for large-scale scraping, offers built-in support for handling requests, follow links, and extract data.
  3. **Weaknesses**: Steeper learning curve compared to BeautifulSoup, more complex setup.

**Selenium**:

* 1. **Overview**: A tool for automating web browsers.
  2. **Strengths**: Can handle JavaScript-rendered content, mimics real user behavior, suitable for scraping dynamic content.
  3. **Weaknesses**: Slower compared to other libraries as it involves browser automation, higher resource consumption.

## Design

### Features of the Scraper

**Product Details Extraction**:

* 1. Product Name
  2. Brand
  3. Price
  4. Discount
  5. Product Description

**Ratings and Reviews**:

* 1. Overall Rating
  2. Number of Ratings
  3. Number of Reviews
  4. Individual Review Content

**Data Storage**:

* 1. Structured format such as CSV or JSON for easy analysis and processing.

### Design Plans

**Navigating Flipkart**:

* 1. Use Selenium for navigating the website due to dynamic content.
  2. Automate search functionality to find products of interest.

**Scraping Data**:

* 1. Use BeautifulSoup to parse the HTML content fetched by Selenium.
  2. Extract relevant data fields and handle missing or inconsistent data.

**Handling Issues**:

* 1. Implement error handling for network issues, captcha, or anti-scraping measures.
  2. Use rotating proxies and user-agents to avoid detection.
  3. Implement retry mechanisms for failed requests.

### Handling Issues

**Website Changes**:

* 1. Regularly update the scraper to adapt to any changes in Flipkart's HTML structure.

**Anti-Scraping Measures**:

* 1. Use rotating proxies and random user-agents.
  2. Implement delays between requests to mimic human behavior.

**Data Consistency**:

* 1. Validate extracted data to ensure consistency and accuracy.

## Testing

### Testing Scenarios

1. **Basic Functionality**: Ensure the scraper can extract product details, prices, and ratings for a single product.
2. **Multiple Products**: Test the scraper's ability to handle multiple products and pages.
3. **Error Handling**: Simulate network issues, captchas, and other potential errors to test robustness.
4. **Performance**: Evaluate the scraper's performance in terms of speed and resource usage.

FLOW CHART

ACCESS PAGE

STORE DATA IN CSV

points

DATA VALIDATION

ERROR HANDLING

EXTRACT PRODUCT LIST

NAVIGATE PRODUCT CATEGORY