**Prompt:**   
  
Create a complete Python web scraper for extracting product information from [WEBSITE\_URL]. The scraper should:

1. Use Selenium and BeautifulSoup for extracting data from interactive product pages

2. Implement a robust deduplication system to prevent duplicate products

3. Save data in three formats: JSON, Excel, and CSV

4. Include error handling and detailed logging

5. Feature parallel processing to increase performance

6. Include resume capability to continue interrupted scraping jobs

The website has the following structure:

- Products are listed on pages with pagination

- Each product must be clicked to view its details

- There are approximately [NUMBER] products across [NUMBER] pages

- Each product has these attributes: [LIST ATTRIBUTES like name, brand, price, description, etc.]

The scraper should include these advanced features:

- Tracking system to remember already processed products

- Anti-blocking measures to respect website terms

- Configurable speed/delay settings to avoid overloading the server

- Intelligent retry mechanism for failed requests

- Memory-efficient data storage for large datasets

- Progress reporting and estimated completion time

For optimal performance, configure the scraper to use:

- Headless browser mode

- Minimal resource loading (disable images, etc.)

- Proper waits and timeouts

- Text parsing optimization for extracting structured data from descriptions

Please provide the complete code with detailed comments explaining each significant section.

**Note:**  
You can customize this template with specific details about any website you want to scrape. For example:

1. For the Natrue.org website, you would add:
   * The specific URL structure
   * Details about the product dialog and its HTML structure
   * Mention the certification levels and other unique data fields
2. For other websites, adjust based on:
   * How products are displayed (cards, lists, grids)
   * Whether JavaScript is required to load content
   * Any login/authentication requirements
   * Special handling for text parsing if needed