Web Scraping Report: Daraz Smartphone Data Collection

Overview

This code implements an automated web scraping solution to collect smartphone product data from Daraz Bangladesh using the crawl4ai library. The script extracts product information and stores it in a MongoDB database.

Key Components

1. Data Model

```python
class Product(BaseModel):
    product_title: str = Field(..., description="Name of the product")
    current_price: str = Field(..., description="Current price in BDT (e.g., '৳ 12,399')")
    product_url: str = Field(..., description="Product detail page URL")
    is_free_delivery: bool = Field(default=False, description="Whether delivery is free")
```

- Uses Pydantic for data validation
- Ensures consistent data structure
- Defines required fields and types

2. Crawler Configuration

- Uses AsyncWebCrawler for asynchronous operation
- Implements page scrolling (6 rounds)
- Sets viewport dimensions (1366x2000)
- Uses headless browser mode

3. Data Extraction Methods

- Primary Method (JavaScript)
    1. Executes client-side JavaScript to:
    2. Handle dynamic content loading
    3. Dismiss popups
    4. Extract product information
    5. Detect pagination status
- Fallback Method (BeautifulSoup)
    1. Parses static HTML when JavaScript extraction fails
    2. Uses multiple CSS selectors for redundancy
    3. Extracts same data points as JavaScript method

4. Data Processing Pipeline

1. Extraction

    - Fetches raw HTML
    - Executes JavaScript
    - Parses DOM structure

2. Normalization

    - Converts relative URLs to absolute
    - Standardizes price format
    - Handles missing values

3. Validation

    - Validates data against Pydantic model
    - Removes invalid entries
    - Deduplicates by URL

4. Storage

    - Stores in MongoDB collection "products2"
    - Maintains unique entries
    - Tracks document count

Performance Features

- Async operation for better throughput
- Random delays between requests (2-4 seconds)
- Configurable scroll rounds and wait times
- Page limit cap (10 pages)

Output and Logging

- Saves HTML and Markdown for debugging
- Reports extraction statistics
- Shows validation results
- Confirms database operations

Key Libraries

- crawl4ai
- BeautifulSoup4
- Pydantic
- PyMongo
- aiohttp

Mongodb Link : Products