iPhone Price Comparison App - Documentation

Execution Flow

Execution Flow (Step-by-Step)

1. Celery Beat runs every 5 minutes and enqueues a task run_all_spiders() into Redis.

2. Celery Worker picks up the task from Redis and executes the following commands:
   - scrapy crawl amazon
   - scrapy crawl flipkart

3. Each spider scrapes model name and price and calls insert_price().

4. insert_price() stores records in MongoDB (if 'iphone' is found in the model name).

5. Flask app serves this data via:
   - /compare/<model_name> endpoint.

Setup Instructions

Setup Instructions

1. Create Project Directory and Virtual Environment:
   mkdir iphone_price_scraper && cd iphone_price_scraper
   python -m venv venv
   venv\Scripts\activate

2. Install Dependencies:
   pip install -r requirements.txt

3. Start Redis and MongoDB:
   Use Docker Desktop or native installation.
   docker-compose up -d

4. Run Spiders Manually:
   scrapy crawl amazon
   scrapy crawl flipkart

5. Start Application:
   Terminal 1: celery -A celery_tasks.tasks worker --loglevel=info --pool=solo
   Terminal 2: celery -A celery_app.app beat --loglevel=info
   Terminal 3: python flask_app.py

6. Access the API:
   http://localhost:5000/compare/iphone

Project Structure

Project Structure

iphone_price_scraper/
|-- celery_app.py

iPhone Price Comparison App - Documentation

```
|-- celery_tasks/tasks.py
|-- config.py
|-- database/insert.py
|-- flask_app.py
|-- requirements.txt
|-- scraper/
    |-- settings.py
    |-- spiders/
        |-- amazon.py
        |-- flipkart.py
```

config.py

```
# config.py
REDIS_BROKER_URL = 'redis://localhost:6379/0'
MONGO_URI = 'mongodb://localhost:27017'
DB_NAME = 'iphone_prices'
```

celery_app.py

```
# celery_app.py
from celery import Celery
import config

app = Celery('tasks', broker=config.REDIS_BROKER_URL)
app.conf.beat_schedule = {
    'scrape-every-5-mins': {
        'task': 'celery_tasks.tasks.run_all_spiders',
        'schedule': 60.0 * 5,
    },
}
```

tasks.py

```
# celery_tasks/tasks.py
from celery import Celery
import subprocess
import config

app = Celery('tasks', broker=config.REDIS_BROKER_URL)

@app.task
def run_all_spiders():
    subprocess.run(['scrapy', 'crawl', 'amazon'])
    subprocess.run(['scrapy', 'crawl', 'flipkart'])
```

insert.py

```
# database/insert.py
from pymongo import MongoClient
import config
```

iPhone Price Comparison App - Documentation

```
client = MongoClient(config.MONGO_URI)
db = client[config.DB_NAME]

def insert_price(data):
    if 'iphone' in data['model_name'].lower():
        db.prices.insert_one(data)
```

flask_app.py

```
# flask_app.py
from flask import Flask, jsonify
from pymongo import MongoClient
import config

app = Flask(__name__)
client = MongoClient(config.MONGO_URI)
db = client[config.DB_NAME]

@app.route('/compare/<model_name>')
def compare(model_name):
    records = list(db.prices.find({'model_name': {'$regex': model_name, '$options': 'i'}}, {'_id':
0}))
    return jsonify(records)

@app.route('/')
def home():
    return "<h1>iPhone Price Comparison API</h1><p>Use /compare/<model_name> to get prices.</p>"

if __name__ == '__main__':
    app.run(debug=True)
```

scrapy.cfg

```
# scrapy.cfg
[settings]
default = scraper.settings
```

settings.py

```
# scraper/settings.py
BOT_NAME = 'scraper'
SPIDER_MODULES = ['scraper.spiders']
NEWSPIDER_MODULE = 'scraper.spiders'
ROBOTSTXT_OBEY = False
```

amazon.py

```
# scraper/spiders/amazon.py
import scrapy
from database.insert import insert_price

class AmazonSpider(scrapy.Spider):
```

```
    name = 'amazon'
    allowed_domains = ['amazon.in']
    start_urls = ['https://www.amazon.in/s?k=iphone']

    def parse(self, response):
        for product in response.css('.s-result-item'):
            model = product.css('h2 span::text').get()
            price = product.css('.a-price-whole::text').get()
            if model and price:
                      insert_price({'model_name': model.strip(), 'price': price.strip(), 'site':
'Amazon'})
```

flipkart.py

```
# scraper/spiders/flipkart.py
import scrapy
from database.insert import insert_price

class FlipkartSpider(scrapy.Spider):
    name = 'flipkart'
    allowed_domains = ['flipkart.com']
    start_urls = []

    def start_requests(self):
        yield scrapy.Request(
            url='https://www.flipkart.com/search?q=iphone',
            headers={
                      'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36'
            }
        )

    def parse(self, response):
        for product in response.css('div._1AtVbE'):
            model = product.css('div._4rR01T::text').get()
            price = product.css('div._30jeq3::text').get()
            if model and price:
                insert_price({
                    'model_name': model.strip(),
                    'price': price.replace('', '').replace(',', '').strip(),
                    'site': 'Flipkart'
                })
```