Web Scraping Uygulaması

Projenin Amacı : Python kullanarak belirlenen kitap alışveriş sitelerinden Python ile ilgili kitapların bilgilerini kazımaktır (web scraping).

İstenilen kriterlere ek olarak;

- Her gün saat 12:00 da sistemin database verisini güncellemesi sağlanmıştır.
- Pandas kütüphanesi kullanılarak iki site arasında fiyat karşılaştırılması yapılmıştır.
- Pandas kütüphanesi kullanılarak aynı yayımcıya ait veya aynı yazara ait kitapların gruplama işlemleri yapılmıştır.

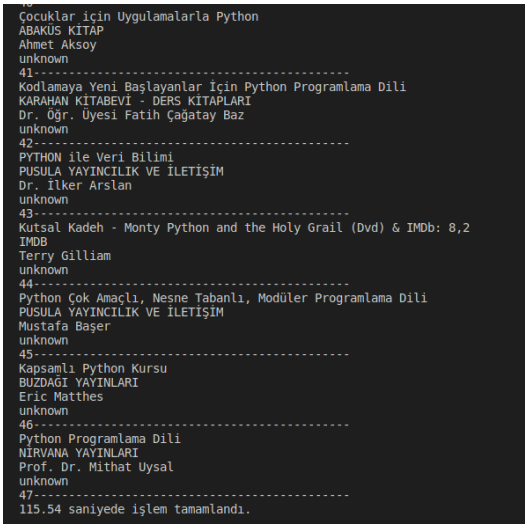
Kullanılan kütüphaneler

1) **selenium :** Selenium, web tarayıcı otomasyonu için kullanılan bir Python kütüphanesidir. Web tabanlı test otomasyonunda veya web veri kazıma (web scraping) gibi senaryolarda kullanılır.
2) **time :** Bu modül zamanla alakalı çeşitli fonksiyonlar sağlar.
3) **pandas :** Python'da veri yükleme, veri ön işleme ve veri temizleme gibi işlemler için genel olarak kullanılan kütüphanedir.
4) **pymongo :** MongoDB sunucusuna bağlanmak, verileri sorgulamak ve değiştirmek ve çeşitli yönetim görevlerini gerçekleştirmek için bir API sağlayarak Python uygulamalarının MongoDB ile etkileşime girmesine izin verir.
5) **datetime :** Zaman, saat ve tarihlerle ilgili işlemler yapmamızı sağlayan önemli bir standart kütüphane modülüdür.
6) **apscheduler :** Python'da zaman tabanlı görevlerin planlanması ve yönetimi için kullanılan bir kütüphanedir. Farklı zamanlama mekanizmalarıyla (örneğin, belirli bir tarih ve saat, tekrarlayan aralıklar, Cron tablosu gibi) planlanabilen görevleri çalıştırır.

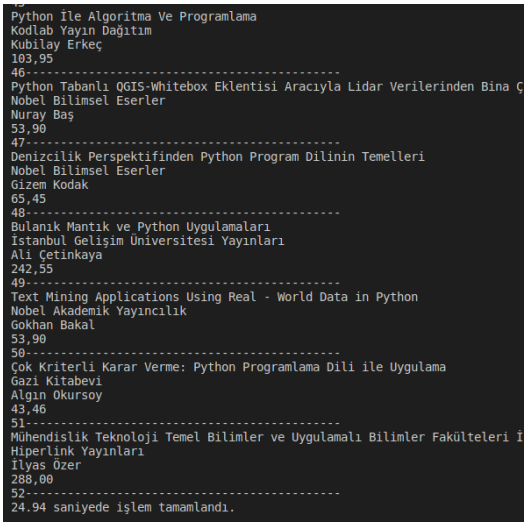
Projenin Yürütülme Esasları ve Aşamaları

➢ **Proje Oluşturulurken Yapılan İşlemler**

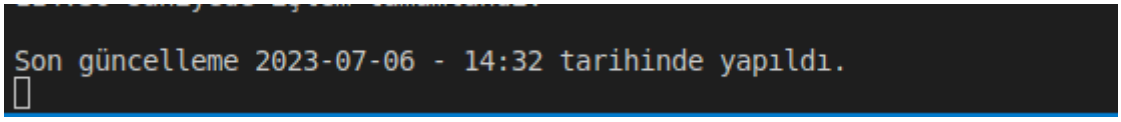
- Kullanılacak olan kütüphane importları yapıldı.
- app.py içerisinde bulunan 145. satırda Bookstore classından bir "app" nesnesi oluşturularak program başlatılmıştır. Ardından kitapyurdu ve kitapsepeti fonksiyonları sırasıyla çağırılmıştır.
- Bookstore adında bir class oluşturup, webdriver ve mongodb bağlantısı oluşturuldu.
- "bookstores" isimli database tanımlanmasının ardından, Studio 3T GUI si kullanılarak, uri bağlantısı yapıldı. Database ve collectionlar oluşturuldu.
- Oluşturulan collectionların daha sonra pandas kütüphanesiyle kullanılabilmesi için csv dosya formatına dönüştüren fonksiyon yazıldı.
- "https://www.kitapyurdu.com/" sitesi üzerinde scraping işlemlerini yapacak olan kitapyurdu adlı fonksiyon oluşturuldu. Chromium web driver yardımıyla siteye erişildi. Searchbox'a otomatik olarak "python" keywordu girilmesi sağlanarak, pythonla ilişkili olan kitaplara erişildi. Verilerin sınırlı sayıda olduğu göz önünde bulundurarak "select box"dan seçim yapılarak ilerlenmiştir. CSS (XPath, class name, id, vb. element niteleyicileri) yardımıyla istenilen kitap bilgilerine erişildi. Price değişkeni için old ve new price varyasyonları olduğundan koşullandırma yapılarak kitap bilgileri print edildi. Verilerin database'e aktarılması sırasında price bilgisi olmayan veriler kayıt dışı tutulmuştur. Fonksiyonun çalışmayı tamamlamasına kadar geçen süre yazdırılmıştır. Fonksiyonda son olarak, oluşturulan collection "collection_to_csv" fonksiyonuna gönderilmiştir.



- "https://www.kitapsepeti.com/" sitesi üzerinde de benzer şekilde scraping işlemleri gerçekleştirilmiştir. Site şablonunda bulunan farklılıklardan dolayı ilave olarak "Satıştakiler" checkbox'ı yardımıyla price conditionları oluşturulmadan ilgili kitap bilgileri veritabanına aktarılmıştır.

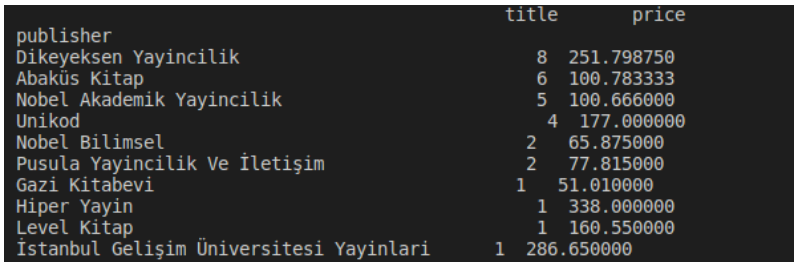


- Belirlenen gün ve saatler için güncelleme yapacak olan "updates" fonksiyonu oluşturulmuş ve işlem tamamlandığında terminale "güncelleme yapıldı" çıktısı print edilmiştir.



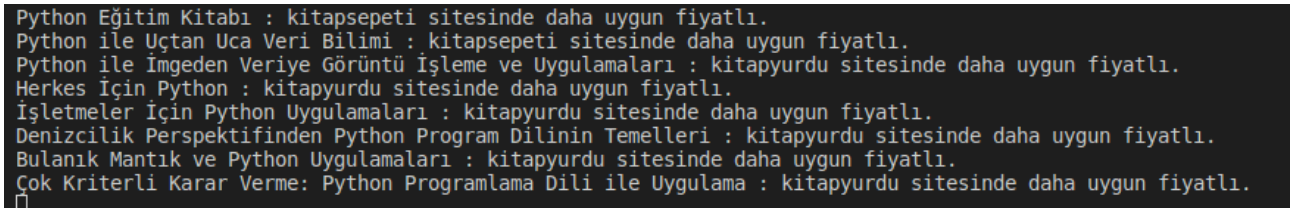
- Pandas kütüphanesi yardımıyla oluşturulan csv dosyaları dataframe olarak sisteme alındı.
- "df_prepare" fonksiyonunda; id columnu drop edilmesi, price columnunun float tipine çevrilmesi ve publisher columnundaki stringlerin camel case formatına gelmesi sağlanmıştır. Dataframe düzenleme işlemlerini defaten yapmamak adına fonksiyonlaştırılmıştır.

- “groupby_func_writer_or_publisher “ fonksiyonu kullanılarak verilen dataframe ve sütun(writer ve publisher) değerleri için gruplama yapılarak, o gruba ait kitap sayısına ve price ortalamalarına erişilebilmektedir.



publisher	title	price
Dikeyeksen Yayincilik	8	251.798750
Abaküs Kitap	6	100.783333
Nobel Akademik Yayincilik	5	100.666000
Unikod	4	177.000000
Nobel Bilimsel	2	65.875000
Pusula Yayincilik Ve İletişim	2	77.815000
Gazi Kitabevi	1	51.010000
Hiper Yayin	1	338.000000
Level Kitap	1	160.550000
İstanbul Gelişim Üniversitesi Yayinlari	1	286.650000

- “compare_prices” fonksiyonu kullanılmadan önce iki dataframe ortak titlelar üzerinden merge edilmiştir. Sonrasında bu fonksiyon yardımıyla aynı isim ve yazara sahip kitapların fiyat karşılaştırması yapılarak sonuç print edilmiştir.



```
Python Eğitim Kitabı : kitapsepeti sitesinde daha uygun fiyatlı.
Python ile Uçtan Uca Veri Bilimi : kitapsepeti sitesinde daha uygun fiyatlı.
Python ile İmgeden Veriye Görüntü İşleme ve Uygulamaları : kitapyurdu sitesinde daha uygun fiyatlı.
Herkes İçin Python : kitapyurdu sitesinde daha uygun fiyatlı.
İşletmeler İçin Python Uygulamaları : kitapyurdu sitesinde daha uygun fiyatlı.
Denizcilik Perspektifinden Python Program Dilinin Temelleri : kitapyurdu sitesinde daha uygun fiyatlı.
Bulanık Mantık ve Python Uygulamaları : kitapyurdu sitesinde daha uygun fiyatlı.
Çok Kriterli Karar Verme: Python Programlama Dili ile Uygulama : kitapyurdu sitesinde daha uygun fiyatlı.
```

➢ Oluşturulan Projenin Ayağa Kaldırılması İşlemleri

- Kullanılacak olan işletim sistemi ve tarayıcı sürümüne uygun driverin yüklenmesi. Uygulama ubuntu işletim sistemi üzerinde geliştirilmiş olup, chrome webdriver kullanılmıştır. Takip eden adımlar linux işletim sistemine göredir.
- Terminalden env dosyası “python3 -m venv env” komutuyla oluşturulur.
- “source venv/bin/activate” komutuyla virtual environment active edilir.
- “pip3 install -r requirements.txt” dosyası yardımıyla gerekli kütüphanelerin kurulumu sağlanır.
- “python3 app.py” komutu girilerek uygulama başlatılır.