



MERSİN ÜNİVERSİTESİ
BİLİŞİM SİSTEMLERİ VE TEKNOLOJİLERİ BÖLÜMÜ

MATEMATİK YAZILIMLARI

2023-2024 BAHAR DÖNEMİ

Proje Adı	Otoişlem
Dersin Öğretim Üyesi	Hüseyin Yanık
Projeyi Yapan	Simge Gündoğdu
Proje Teslim Tarihi	17.06.2024

• GİRİŞ

Proje, beş farklı konsol otomasyonundan oluşmaktadır. Projenin genel amacı, günlük hayatta ve bilgisayar kullanımında yapacağımız işlemleri kolaylaştırmaktır.

1-Alışveriş Listesi Otomasyonu: Evdeki verilere göre alışveriş listesini daha kolay oluşturmayı ve takibini yapmayı amaçlar.

2-E-posta Gönderme Otomasyonu: Düzenli olarak e-posta ihtiyacı duyulan zamanları otomatikleştirerek zamandan tasarruf sağlamayı hedefler.

3-Metin Analizi Otomasyonu: Metinleri detaylıca incelemek yerine, metindeki istatistikleri çıkartarak zamandan tasarruf sağlar.

4-Otomatik Dosya Yedekleme Otomasyonu: Kaybolmasından endişe edilen dosyaları belli aralıklarla yedekleyerek olası veri kaybının önüne geçmeyi amaçlar.

5-PDF Birleştirme Otomasyonu: Seçilen PDF dosyalarını kolay bir şekilde birleştirmeyi hedefler

• TEORİ

Alışveriş otomasyonu; Projenin önemi, günlük hayatta alışveriş ve stok yönetiminin düzenli bir şekilde yapılmasını sağlamaktır. Bu, özellikle büyük aileler, işletmeler veya benzeri kuruluşlar için zaman ve maliyet tasarrufu sağlar. Kodun işlevselliği, kullanıcı dostu bir menü aracılığıyla sunulmakta ve kullanıcıya stok güncellemesi yapma veya eksik ürünleri tespit edip alışveriş listesi oluşturma seçenekleri sunmaktadır. Kullanıcıya interaktif bir deneyim sunarak, işlemlerin kolay ve anlaşılır bir şekilde gerçekleştirilmesi sağlanmaktadır.

E-posta gönderme; bu Python betiği, kullanıcıdan alınan bilgilerle Gmail üzerinden e-posta göndermeyi sağlar.

Metin analizi;

Bu Python betiği, kullanıcı tarafından belirtilen metin dosyasının istatistiklerini hesaplamak için tasarlanmıştır. Metin dosyasında karakter sayısını, kelime sayısını ve cümle sayısını bulur.

Dosya yedekleme;

Bu Python betiği, dosya yönetimi ve veri koruma açısından önemli bir işlevi yerine getirir: belirtilen kaynak dizinindeki dosyaların hedef dizine yedeklenmesini sağlar. Veri kaybını önlemek ve dosya yönetimini kolaylaştırmak amacıyla kullanıcı dostu bir arayüz sunar.

PDF birleştirme; PDF birleştirme işlemi, birden fazla belgenin tek bir dosya içinde toplanmasını sağlayarak, doküman yönetimindeki düzeni ve erişimi iyileştirir. Özellikle iş ortamlarında veya akademik çalışmalarda, belge bütünlüğünün korunması ve paylaşımının kolaylaştırılması, verimlilik ve işbirliği açısından önemlidir.

• MATERYAL YÖNTEM

Alışveriş listesi otomasyonu

```
D:\otomasyon\pythonProject4\.venv\Scripts\python.exe D:\otomasyon\pythonProject4\AlisVerisListesi_otomasyon\main.py
1. Stokları Güncelle
2. Alışveriş Listesi Oluştur
3. Çıkış
Seçiminizi yapın: 1
Stok güncellemesi
süt için yeni stok miktarını girin (mevcut: 2): 3
yumurta için yeni stok miktarını girin (mevcut: 10): 5
ekmek için yeni stok miktarını girin (mevcut: 1): 0
elma için yeni stok miktarını girin (mevcut: 5): 2
1. Stokları Güncelle
2. Alışveriş Listesi Oluştur
3. Çıkış
Seçiminizi yapın:
```

Kullanılan Yöntemler ve Yazılımlar

1. ****JSON Kütüphanesi (`json`)**:**

- `json` kütüphanesi, Python'da JSON formatında veri işlemek için kullanılır. `json.dump()` fonksiyonu, Python veri yapılarını JSON formatına dönüştürerek belirtilen dosyaya yazar.

2. ****Datetime Modülü (`datetime`)**:**

- `datetime` modülü, tarih ve saat manipülasyonları için kullanılır. `datetime.now()` metodu, şu anki tarih ve saati alır ve belirtilen formatla biçimlendirir (`strftime('%Y-%m-%d')`).

3. ****Kullanıcı Girdisi Alma (`input()`) fonksiyonu**:**

- `input()` fonksiyonu, kullanıcıdan metin tabanlı girdiler almak için kullanılır. Bu projede, kullanıcıdan stok miktarı güncellemeleri ve menü seçimleri için kullanılır.

4. ****Döngüler ve Koşullu İfadeler**:**

- `for` döngüsü, stok güncelleme işlemlerinde ve alışveriş listesi oluşturma işlemlerinde kullanılır.
- `if` koşullu ifadeleri, stokların yetersiz olduğu durumları kontrol ederek alışveriş listesi oluşturma sürecini yönetir.

Projenin Önemi ve Faydaları

Bu proje, küçük ölçekli bir işletmenin veya bireyin envanter yönetimini kolaylaştırır. Özellikle ev kullanıcıları için, hangi ürünlerin stokta olduğunu görmek ve hangi ürünlerin eksik olduğunu anlamak büyük önem taşır. Bu yönetim sistemi sayesinde kullanıcılar:

- ****Stok Güncellemesi:**** Kullanıcılar, her ürün için mevcut stok miktarını güncelleyebilir, böylece gerçek zamanlı envanter takibi yapabilirler.

- Alışveriş Listesi Oluşturma:Günlük ihtiyaçlar ile mevcut stokları karşılaştırarak eksik olan ürünleri belirleyip, bu ürünleri içeren bir alışveriş listesi otomatik olarak oluşturulur. Bu, kullanıcılara alışveriş sırasında zaman kazandırır ve gereksiz alımların önüne geçer.

stok_guncelle() Fonksiyonu

Bu fonksiyon, kullanıcıya her bir ürün için stok güncellemesi yapma imkanı verir. Kullanıcıdan alınan yeni stok miktarı, doğrulanarak güncellenir.

def stok_guncelle():

```
    print("Stok güncellemesi")
```

```
    for urun in stoklar.keys():
```

```
        yeni_stok = input(f"{urun} için yeni stok miktarını girin (mevcut: {stoklar[urun]}): ")
```

```
        if yeni_stok.isdigit():
```

```
            stoklar[urun] = int(yeni_stok)
```

E-POSTA GÖNDERME

```
4
5
6  ▶ if __name__ == "__main__":
7      gonderen = input("Gönderen e-posta adresini girin: ")
8      sifre = input("Gönderen e-posta şifresini girin: ")
9      alici = input("Alıcı e-posta adresini girin: ")
0      konu = input("E-posta konusunu girin: ")
1      icerik = input("E-posta içeriğini girin: ")
2
3      email_gonder(gonderen, sifre, alici, konu, icerik)
4
```

Kullanılan Yöntemler ve Yazılımlar

smtplib Kütüphanesi: E-posta göndermek için SMTP (Simple Mail Transfer Protocol) üzerinden iletişim kurar.

email.mime Modülleri: E-posta mesajlarının MIME (Multipurpose Internet Mail Extensions) formatında oluşturulmasını sağlar.

email_gonder() Fonksiyonu

Bu fonksiyon, kullanıcının sağladığı bilgilerle bir e-posta gönderir. Eğer işlem başarılı olursa gönderilen adresi bildirir, aksi halde bir hata mesajı gösterir.

METİN ANALİZ

Bu Python betiği, belirli bir metin dosyasının içeriğini analiz ederek metin istatistikleri hesaplar. Kullanılan yöntemler ve yazılımlar şunlardır:

Kullanılan Yöntemler ve Yazılımlar

-Python Dosya İşlemler: `'open()'` fonksiyonuyla dosya okuma işlemi yapılır.

String İşlemleri: Metin manipülasyonu ve işlemleri için Python'un yerleşik string fonksiyonları kullanılır.

-Dosya Kontrolü: `'os.path.isfile()'` fonksiyonu ile dosya varlığı kontrol edilir.

-Hata Yönetimi: `'try-except'` yapısı ile dosya bulunamama durumu ele alınır.

`'metin_istatistikleri_hesapla()'` Fonksiyonu

Bu fonksiyon, bir metin dosyasının yolunu parametre olarak alır ve şu istatistikleri hesaplar:

Karakter Sayısı: Metnin içindeki toplam karakter sayısını hesaplar (`'len(metin)'`).

Kelime Sayısı: Metindeki toplam kelime sayısını hesaplar (`'len(metin.split())'`).

Cümle Sayısı: Metindeki toplam cümle sayısını hesaplar (nokta, ünlem, soru işareti ile bitenler için `'sum(metin.count(c) for c in ".!?"")'`).

Ana Program (`'__main__'`)

Ana program kullanıcıdan bir metin dosyası yolunu alır (`'input()'` fonksiyonuyla) ve bu dosyanın varlığını kontrol eder (`'os.path.isfile()'`). Dosya mevcutsa `'metin_istatistikleri_hesapla()'` fonksiyonunu çağırarak istatistikleri ekrana yazdırır. Dosya mevcut değilse geçersiz dosya yolu mesajı verir.

Bu şekilde, betik dosya işlemleri, string manipülasyonları ve hata yönetimi kullanarak metin istatistiklerini basit ve etkili bir şekilde hesaplar ve kullanıcıya sunar.

DOSYA YEDEKLEME

Bu Python betiği, belirtilen bir kaynak dizinindeki dosyaları hedef bir dizine yedekler. Kullanılan yöntemler ve yazılımlar şunlardır:

Kullanılan Yöntemler ve Yazılımlar

- Python ``os`` Modülü: Dizin işlemleri için kullanılır (``os.path.join()``, ``os.makedirs()``, ``os.listdir()``, ``os.path.isfile()``).
- Python ``shutil`` Modülü: Dosya ve dizin kopyalama işlemleri için kullanılır (``shutil.copy2()``).
- Zaman Modülü: Yedeklemenin yapıldığı zaman bilgisini almak için ``time.strftime()`` kullanılır.

``dosya_yedekle()`` Fonksiyonu

Bu fonksiyon, belirtilen bir kaynak dizinindeki dosyaları hedef bir dizine yedekler. İşlemler şu adımlarla gerçekleştirilir:

1. Yedekleme Tarihi: ``time.strftime('%Y-%m-%d_%H-%M-%S')`` ile geçerli zaman bilgisini alarak yedekleme tarihini oluşturur.
2. Hedef Yedek Dizinini Oluştur: ``os.makedirs()`` ile hedef yedek dizinini oluşturur.
3. Dosyaları Yedekleme: ``os.listdir()`` ile kaynak dizinindeki tüm dosyaları döngüye alır. Her dosya için:
 - Dosya yolunu belirler (``os.path.join()``).
 - Dosyanın bir dosya olup olmadığını kontrol eder (``os.path.isfile()``).
 - Dosyayı ``shutil.copy2()`` ile hedef yedek dizinine kopyalar ve işlemi ekrana yazdırır.
4. Yedekleme Tamamlandı: Yedekleme işlemi başarıyla tamamlandıktan sonra, yedekleme dizinini ekrana yazdırır.
5. Hata Yönetimi: Herhangi bir hata durumunda (örneğin, dizin oluşturma hatası veya dosya kopyalama hatası), ``try-except`` bloğu ile hata mesajını ekrana yazdırır.

Ana Program (``__main__``)

Ana program kullanıcıdan kaynak ve hedef dizinlerini alır (`input()` fonksiyonuyla) ve bu dizinlerin varlığını kontrol eder (`os.path.isdir()`). Her iki dizin de mevcut ise `dosya_yedekle()` fonksiyonunu çağırarak yedekleme işlemini başlatır. Dizinlerden herhangi biri geçersiz ise kullanıcıya geçersiz dizin yolu mesajı verir.

Bu betik, dosya işlemleri ve hedefe yönelik yedekleme işlemlerini otomatikleştirerek kullanıcıya basit bir arayüz sunar ve dosyaların güvenli bir şekilde yedeklenmesini sağlar.

PDF BİRLEŞTİRME

Bu Python betiği, belirtilen bir dizinde bulunan PDF dosyalarını birleştirip birleşik bir PDF dosyası oluşturur. İşlem aşamaları ve kullanılan yöntemler şu şekildedir:

Kullanılan Yöntemler ve Yazılımlar

- Python `os` Modülü: Dizin işlemleri için kullanılır (`os.listdir()`, `os.path.join()`).
- PyPDF2 Kütüphanesi: PDF dosyalarını birleştirmek için kullanılır (`PyPDF2.PdfFileMerger()`).
- ****PDF Sıkıştırma Araçları****: Sıkıştırma işlemi için bu örnekte doğrudan kullanılmamıştır, ancak genellikle sıkıştırma için `PyPDF2` dışında `PyMuPDF`, `ReportLab` veya `ghostscript` gibi kütüphaneler/araçlar tercih edilebilir.

`pdf_birlestir_sikistir()` Fonksiyonu

Bu fonksiyon, belirtilen dizindeki PDF dosyalarını birleştirir ve isteğe bağlı olarak sıkıştırma işlemi yapar.

1. PDF Dosyalarını Listeleme: `os.listdir()` ve list comprehension kullanılarak belirtilen dizindeki tüm PDF dosyaları `pdf_listesi` listesine eklenir.

2. PDF Birleştirme: `PyPDF2.PdfFileMerger()` ile yeni bir PDF birleştirme nesnesi oluşturulur (`birlesik_pdf`). Daha sonra `birlesik_pdf.append(pdf_yol)` ile tüm PDF dosyaları birleştirme işlemine eklenir.

3. Birleştirilmiş PDF Oluşturma: Birleştirme işlemi tamamlandıktan sonra `birlesik_pdf_yolu` adlı bir dosya oluşturularak bu birleştirilmiş PDF dosyası `wb` (write binary) modunda açılır ve `birlesik_pdf.write(dosya)` ile içeriği bu dosyaya yazılır.

4. Bilgilendirme Mesajı: İşlem başarıyla tamamlandığında, oluşturulan birleştirilmiş PDF dosyasının yolunu ekrana yazdırır.

Ana Program (`__main__`)

- Kullanıcıdan `input()` ile bir dizin yolu alınır (`dizin`).

- `os.path.isdir(dizin)` ile bu dizinin varlığı kontrol edilir. Eğer varsa `pdf_birlestir_sikistir(dizin)` fonksiyonu çağrılır, aksi takdirde geçersiz dizin yolu uyarısı verilir.

Bu betik, birden fazla PDF dosyasını tek bir dosya haline getirmek için kullanıcıya basit bir arayüz sunar ve işlemleri otomatikleştirir.

• SONUÇLAR VE DEĞERLENDİRME

1-Alışveriş Listesi

Amaç: Günlük ihtiyaçlara göre stokları güncelleyip, eksik ürünleri belirleyerek alışveriş listesi oluşturmak.

Kazanımlar:

Stok Güncelleme: Kullanıcıya mevcut stokları güncelleme imkanı sağlandı. Kullanıcı, her bir ürün için yeni stok miktarını girebilir.

Alışveriş Listesi Oluşturma: Günlük ihtiyaçlar ile mevcut stoklar karşılaştırılarak eksik ürünler belirlendi. Bu eksik ürünler, JSON formatında alışveriş listesi olarak kaydedildi.

Kullanılan yöntemler: json modülü ile veriler JSON formatında kaydedildi, datetime modülü ile tarih bilgisi alındı, kullanıcı arayüzü için basit bir menü oluşturuldu.

2- E-posta gönderme

Bu projede Python kullanarak güvenli bir şekilde e-posta gönderme işlemi gerçekleştirilmiştir. Elde edilen sonuçlar ve kazanımlar aşağıda özetlenmiştir:

Proje Hedefleri ve Sonuçları

E-posta Gönderebilme Yeteneği:

smtp lib kullanılarak Gmail SMTP sunucusuna bağlandı.

TLS güvenliği kullanılarak bağlantı sağlandı.

Kullanıcı girişi yapılarak doğrulama gerçekleştirildi.

MIME kullanılarak e-posta içeriği oluşturuldu ve gönderildi.

Güvenliği Artırmak:

Şifre ve e-posta adresi çevresel değişkenlerle saklandı, güvenlik riskleri azaltıldı.

3-Metin analiz

Proje Hedefleri ve Sonuçları

Hedef 1: Metin Dosyasının Karakter Sayısını Hesaplama

Dosya başarıyla okunarak karakter sayısı hesaplanmıştır.

len(metin) kullanılarak toplam karakter sayısı belirlenmiştir.

Hedef 2: Metin Dosyasının Kelime Sayısını Hesaplama

Metindeki kelimeler split() metodu ile ayrılarak kelime sayısı hesaplanmıştır.

len(kelimeler) ile toplam kelime sayısı belirlenmiştir.

Hedef 3: Metin Dosyasının Cümle Sayısını Hesaplama

Nokta, ünlem ve soru işareti gibi cümle sonlandırıcıları kullanılarak cümle sayısı hesaplanmıştır.

sum([cümle.count(c) for cümle in cümleler for c in cümle_ayrac]) ifadesiyle cümle sayısı belirlenmiştir.

Elde Edilen Kazanımlar

Dosya İşlemleri ve Hata Yönetimi: Dosya okuma işlemleri ve FileNotFoundError yönetimi konusunda deneyim kazanıldı.

Metin İşleme: Karakter, kelime ve cümle sayısı hesaplama gibi temel metin işleme teknikleri uygulandı.

Python Programlama Becerisi: Python dilinde metin işleme ve dosya okuma/yazma konularında pratik yapıldı.

4-Dosya yedekleme

Proje Hedefleri ve Sonuçları

Hedef 1: Dosyaları Yedekleme İşlemini Gerçekleştirme

Program, belirtilen dizindeki tüm dosyaları belirli bir hedef dizine yedekler.

Yedekleme işlemi için `shutil.copy2` fonksiyonu kullanılarak dosyaların tüm meta verileriyle birlikte kopyalanması sağlanır.

Hedef 2: Yedekleme Dizini İçin Zaman Damgası Ekleme

Yedekleme dizini, yedeklemenin yapıldığı tarih ve saat bilgilerini içeren bir isimle oluşturulur.

Bu işlem için `time.strftime` fonksiyonu kullanılarak zaman damgası oluşturulur ve yedek dizinine eklenir.

Elde Edilen Kazanımlar

Dosya İşlemleri ve Hata Yönetimi: Dosya kopyalama ve izin oluşturma işlemleri hakkında deneyim kazanıldı.

Zaman Damgası Kullanımı: Yedekleme işlemlerine tarih ve saat bilgisi eklenerek versiyonlama sağlandı.

Python Programlama Becerisi: Python dilinde dosya yönetimi ve zaman işlemleri konularında pratik yapıldı.

5-PDF Birleştirme

Amaç: PDF birleştirme ve sıkıştırma işlemlerini otomatize etmek.

Kazanımlar:

PDF birleştirme: PyPDF2 kütüphanesi kullanılarak birden fazla PDF dosyası tek bir PDF dosyasında birleştirildi.

Kullanılan yöntemler: `os` modülü ile izin işlemleri yapıldı, PyPDF2 kütüphanesi ile PDF işlemleri gerçekleştirildi.

Kullanılan yazılımlar: Python dilinde `os`, PyPDF2 kütüphaneleri.

Sıkıştırma işlemi: Örnekte doğrudan sıkıştırma işlemi yapılmamıştır, ancak genelde PDF sıkıştırma için ek araçlar kullanılabilir.