### ****1. Projenin Amacı****

**Açıklama:** Bu uygulamanın amacı, Windows Olay Görüntüleyicisi'ndeki olayları belirli bir tarih aralığında listelemek, olayların uyarı seviyelerini görselleştirmek ve bu bilgileri Word formatında kaydetmektir. Proje, elektronik delil dersinde olayların analizini kolaylaştırmak için geliştirilmiştir.

### ****2. Kullanılan Teknolojiler ve Kütüphaneler****

**Python**: Uygulamanın temel programlama dili.

**Tkinter**: Arayüz oluşturma kütüphanesi.

**tkcalendar**: Tarih seçimi için kullanılan bir Tkinter bileşeni.

**matplotlib**: Grafik çizimi için kullanılan kütüphane.

**docx (python-docx)**: Olayları Word belgesine kaydetmek için kullanılan kütüphane.

**win32evtlog**: Windows Olay Görüntüleyicisi'ne bağlanıp olayları almak için kullanılan kütüphane.

### ****3. Uygulamanın Arayüz Yapısı****

#### ****İlk Ekran - Giriş Ekranı****

**Açıklama**: Uygulamanın ilk ekranı “Elektronik Delil Dersi Ödev” başlığına sahiptir ve merkezde bir “Tarih Sorgula” butonu bulunur.

**Arayüz Özelleştirmesi**: Arka plan rengi siyah, metin ve buton yazıları pembe renktedir. Butona gelindiğinde gri renge dönüşmektedir.

#### ****İkinci Ekran - Tarih Sorgulama ve Görselleştirme Ekranı****

**Açıklama**: Tarih seçimi yapılabilen ve olay seviyelerinin grafiğe döküldüğü ekrandır. İki ana buton bulunmaktadır:

**Grafik Oluştur**: Seçilen tarih aralığında oluşan olayları grafikte görselleştirir.

**Word’e Kaydet**: Seçilen tarih aralığında oluşan olayların ayrıntılarını bir Word belgesine kaydeder.

**Arayüz Özelleştirmesi**: Arka plan siyah, metin ve butonlar pembe renkte, butonlara gelindiğinde gri renge döner.

### ****4. Uygulamanın Kod Yapısı****

**Gerekli Kütüphanelerin İçe Aktarılması**

import win32evtlogfrom datetime

import datetimeimport matplotlib.pyplot as pltfrom collections

import Counterfrom docx

import Documentimport tkinter as tkfrom tkinter

import messageboxfrom tkcalendar

import DateEntry

**Açıklama**: Olayların alınması, grafik çizimi, arayüz, tarih seçimi ve Word dosyasına kaydetme işlemleri için gerekli kütüphaneler içe aktarılır.

#### ****Olayları Çekme Fonksiyonu****

def get\_events(start\_date, end\_date):

...

**Açıklama**: Bu fonksiyon, belirtilen tarih aralığındaki olayları Windows Olay Görüntüleyicisi’nden çeker ve bir liste olarak döner. Fonksiyon, start\_date ve end\_date parametrelerini kullanarak yalnızca bu aralıktaki olayları filtreler.

#### ****Grafik Oluşturma Fonksiyonu****

def plot\_event\_levels(events):

...

**Açıklama**: Olayları Error, Warning, ve Information seviyelerinde kategorilere ayırarak bir çubuk grafik oluşturur. matplotlib kütüphanesi kullanılarak olay sayıları görselleştirilir.

#### ****Word’e Kaydetme Fonksiyonu****

def save\_to\_word(events, filename="Event\_Logs.docx"):

...

**Açıklama**: Bu fonksiyon, olayları Word belgesine kaydeder. Her bir olayın EventID, SourceName, EventType, ve Message bilgilerini kaydederek bir rapor oluşturur.

### ****5. Arayüz Kodları****

#### ****Ana Ekran (İlk Ekran)****

root = tk.Tk()

root.configure(bg="black")

...

**Açıklama**: İlk ekran, uygulamayı başlatır ve “Elektronik Delil Dersi Ödev” başlığını ve “Tarih Sorgula” butonunu içerir. Butona basıldığında ikinci ekrana geçilir.

#### ****İkinci Ekran (Tarih Sorgulama ve Grafik Ekranı)****

def create\_main\_screen():

...

**Açıklama**: Bu ekran, olayları seçilen tarih aralığında görselleştirmek ve Word belgesine kaydetmek için tasarlanmıştır. “Grafik Oluştur” ve “Word’e Kaydet” butonlarıyla işlev sağlar.

### ****6. Sonuç ve Değerlendirme****

**Açıklama**: Bu projede, kullanıcıların belirli bir tarih aralığındaki olayları görselleştirebilmesi ve bir rapor olarak kaydedebilmesi sağlanmıştır. Uygulama, arayüz tasarımı ve işlevselliği ile Windows Olay Görüntüleyicisi'ni analiz etmek için pratik bir çözüm sunmaktadır.

### ****7. Ek Notlar****

**Tarih Formatı Uyum Sorunları**: Windows Olay Görüntüleyici tarih formatının Python’un datetime modülüyle uyumsuz olması durumunda hata alınabilir. Bu durumda, tarih formatının uyumluluğunu sağlamak için ek dönüşümler yapılabilir.

**Kütüphane Gereksinimleri**: Uygulamanın düzgün çalışması için tkcalendar, matplotlib, python-docx ve pywin32 kütüphanelerinin yüklü olması gereklidir.

**8.Uygulama Resimleri**















