**QR Kod Oluşturma Projelerinin Karşılaştırması**

**Mevcut Projenin Açıklaması**

Proje, kullanıcıdan alınan veri ve dosya adı doğrulanarak **qrcode** kütüphanesi aracılığıyla QR kodu oluşturur, kaydeder ve GUI üzerinde görüntüler. Kullanıcı deneyimini artırmak için reset ve kapanış onaylama gibi özellikler de bulunmaktadır.

**QR Kodu Oluşturma Süreci**

**1. QRCode Sınıfının Başlatılması**

Öncelikle, QR kodunu oluşturmak için qrcode.QRCode sınıfından bir örnek oluşturulur. Bu örnek, QR kodunun yapılandırmasını belirler.

qr = qrcode.QRCode(version=1, box\_size=6, border=4)

* **version=1:** QR kodunun boyutunu ve veri kapasitesini belirler. Daha yüksek versiyonlar, daha fazla veri içerir.
* **box\_size=6:** QR kodundaki her bir modülün (küçük karelerin) piksel cinsinden boyutunu ayarlar.
* **border=4:** QR kodunun kenarındaki boşlukların sayısını belirler. Bu, kodun taranabilirliğini artırır.

**2. Verinin QR Koduna Eklenmesi**

Kullanıcıdan alınan veri, QR koduna eklenir. qr.add\_data(qrcode\_data)

* **qrcode\_data:** Kullanıcının QR koduna dönüştürmek istediği metin veya URL'dir.

**3. QR Kodunun Oluşturulması**

Verinin QR koduna uygun şekilde yerleştirilmesi için QR kodu oluşturulur. qr.make(fit=True)

* **fit=True:** QR kodunun, eklenen verinin tamamını sığdırabilmesi için gerekli boyutun otomatik olarak ayarlanmasını sağlar.

**4. QR Kodunun Görselleştirilmesi**

Oluşturulan QR kodu, görsel bir formata dönüştürülür.

qrcode\_image = qr.make\_image(fill\_color='black', back\_color='white')

* **fill\_color='black':** QR kodundaki modüllerin (küçük karelerin) rengi.
* **back\_color='white':** QR kodunun arka plan rengi.

**5. QR Kodunun Kaydedilmesi**

Oluşturulan QR kodu, belirlenen dosya adıyla .png formatında kaydedilir.

qrcode\_image.save(name)

* **name:** QR kodunun kaydedileceği dosya adıdır (örneğin, qr\_code.png).

**QR Kod Üretici Projelerinin Karşılaştırması**

Bu bölümde, **1.proje** ve **2.proje** iki farklı Python tabanlı QR Kod Üreteci projesi karşılaştırılmıştır. Her iki proje de Tkinter kullanarak grafiksel kullanıcı arayüzü (GUI) sunmakta olup, QR kodlarını oluşturmak için farklı kütüphaneler kullanmaktadır.

**1. Kullanılan Kütüphaneler**

* **Kode 1:**
  + **tkinter**, **ttk**, **messagebox**, **filedialog**
  + **qrcode**: QR kodu oluşturmak için kullanılır.
* **Kode 2:**
  + **tkinter**, **filedialog**, **messagebox**
  + **segno**: Yüksek hata düzeltme seviyeleri sunar.
  + **PIL** (Pillow): Görüntü işleme ve QR kodunu yüksek kaliteli görüntülemek için kullanılır.

**2. Kullanıcı Arayüzü Tasarımı**

* **Kode 1:**
  + Sekmeli arayüz (ttk.Notebook) ve Canvas kullanılarak widget'lar belirli koordinatlarda yerleştirilmiştir.
  + Özel stil ayarları uygulanmıştır.
* **Kode 2:**
  + Basit ve dikey düzenleme (pack kullanılarak) ile widget'lar sıralanmıştır.
  + Varsayılan Tkinter stilleri kullanılmıştır, daha sade bir görünüm sağlar.

**3. QR Kodunun Oluşturulması**

**Kode 1:**

* **Veri ve Dosya Adının Alınması:** Kullanıcıdan alınan veri ve dosya adı, data\_entry ve filename\_entry alanlarından alınır.
* **Doğrulama:** Dosya adı boşsa hata mesajı gösterilir.
* **Kullanıcı Onayı:** QR kodu oluşturmak için kullanıcıdan onay alınır.
* **QR Kodunun Oluşturulması:** qrcode.QRCode sınıfı kullanılarak QR kodu oluşturulur (version=1, box\_size=6, border=4).
* **Görselleştirme:** Belirtilen renklerle QR kodu görselleştirilir ve kaydedilir.
* **Görüntüleme:** Oluşturulan QR kodu GUI üzerinde görüntülenir.
* **Reset:** Reset butonu aktif hale gelir ve kullanıcı QR kodunu temizleyebilir.

**Kode 2:**

* **Veri Girişi ve Hazırlığı:** Kullanıcıdan alınan URL veya bağlantı, gerekli önekler eklenerek hazırlanır.
* **QR Kodunun Oluşturulması:** segno kütüphanesi kullanılarak yüksek hata düzeltmeli QR kodu oluşturulur (error='H').
* **Görselleştirme ve Kaydetme:** PIL ile QR kodu yüksek kaliteli olarak görselleştirilir ve geçici bir dosyaya kaydedilir.
* **Görüntüleme ve Kaydetme:** QR kodu GUI üzerinde görüntülenir ve kullanıcıya dosya kaydetme imkanı sunulur.

**4. Hata Yönetimi ve Kullanıcı Bildirimleri**

* **Kode 1:**
  + Genel except bloğu kullanılarak tüm hatalar yakalanır, spesifik hata mesajları gösterilmez.
* **Kode 2:**
  + Spesifik Exception yakalama ile detaylı hata mesajları sunulur.

**1. QR Kod Oluşturma Süreci**

* **Kode 1:**
  + **Kullanılan Kütüphane:** qrcode
  + **Süreç:**
    1. Kullanıcıdan QR koduna dönüştürülmek istenen veri (QRcode Data) ve dosya adı (Filename) alınır.
    2. Dosya adı doğrulanır; boş ise hata mesajı gösterilir.
    3. Kullanıcıdan QR kodunu oluşturmak için onay alınır.
    4. qrcode.QRCode sınıfı kullanılarak QR kodu oluşturulur (version=1, box\_size=6, border=4).
    5. Oluşturulan QR kodu belirtilen dosya adıyla .png formatında kaydedilir.
    6. Kaydedilen QR kodu PhotoImage ile yüklenir ve GUI üzerinde görüntülenir.
    7. Reset butonu aktif hale getirilir, böylece kullanıcı QR kodunu temizleyebilir.
    8. Herhangi bir hata durumunda genel bir hata mesajı gösterilir.
* **Kode 2:**
  + **Kullanılan Kütüphane:** segno, PIL (Pillow)
  + **Süreç:**
    1. Kullanıcıdan QR koduna dönüştürülmek istenen URL veya bağlantı alınır ve otomatik olarak "http://" veya "https://" eklenir.
    2. segno kütüphanesi kullanılarak yüksek hata düzeltmeli QR kodu (error='H') oluşturulur.
    3. Oluşturulan QR kodu geçici bir dosyaya (qrcode\_temp.png) kaydedilir.
    4. PIL kullanılarak QR kodu yüklenir ve GUI üzerinde yüksek kaliteli olarak görüntülenir.
    5. Kullanıcıya QR kodunu kaydetme imkanı sunulur (filedialog.asksaveasfilename).
    6. Herhangi bir hata durumunda detaylı hata mesajı gösterilir.

**2. Üstün ve Zayıf Yönler**

* **proje 1:**
  + **Üstün Yönler:**
    - **Kullanıcı Kontrolü:** Kullanıcı, QR kodunu oluştururken dosya adını doğrudan belirleyebilir.
    - **Görselleştirme:** QR kodu doğrudan GUI üzerinde görüntülenir.
    - **Ekstra Özellikler:** Reset butonu ile QR kodunu temizleme imkanı sunar.
  + **Zayıf Yönler:**
    - **Hata Yönetimi:** Genel except bloğu kullanılarak spesifik hatalar yakalanmaz.
    - **Kod Organizasyonu:** GUI ve işlevsellik sıkı entegre edilmiştir, modülerlik düşüktür.
    - **Global Değişken Kullanımı:** Image değişkeni global olarak tanımlanmıştır, bu bellek yönetimini zorlaştırabilir.
* **proje 2:**
  + **Üstün Yönler:**
    - **Yüksek Hata Düzeltme:** segno kütüphanesi ile daha dayanıklı QR kodları oluşturur.
    - **Modülerlik:** Fonksiyonlar daha ayrı ve modüler şekilde tanımlanmıştır.
    - **Kullanıcı Deneyimi:** Otomatik URL önekleme ile kullanıcı hatalarını azaltır.
    - **Detaylı Hata Mesajları:** Spesifik Exception yakalama ile daha açıklayıcı hata mesajları sunar.
  + **Zayıf Yönler:**
    - **Ekstra Özellikler Eksikliği:** Reset butonu yoktur, QR kodunu temizlemek için doğrudan bir seçenek sunulmamıştır.
    - **Stil Özelleştirmeleri:** Varsayılan Tkinter stilleri kullanılmıştır, daha sade bir görünüm sağlar.
    - **Geçici Dosya Kullanımı:** QR kodu geçici dosyada saklanır, dosya yönetimi açısından eksiklikler olabilir.

**Kaynak:**

* [How to Build a GUI QR Code Generator and Reader in Python - The Python Code](https://thepythoncode.com/article/make-a-qr-code-generator-and-reader-tkinter-python#h_88678367311669457000033) .
* [Generate Beautiful QR Codes With Python – Real Python](https://realpython.com/python-generate-qr-code/#using-python-to-generate-a-basic-qr-code) .