Proje 1.

class Arac:

def \_\_init\_\_(self, arac\_id, model, arac\_yili, fiyat, kiralama\_durumu):

self.arac\_id = arac\_id

self.model = model

self.arac\_yili = arac\_yili

self.fiyat = fiyat

self.kiralama\_durumu = kiralama\_durumu

def arac\_durumu\_guncelle(self, durum):

self.kiralama\_durumu = durum

# 5 araç verisi ekle

arac1 = Arac(1, "Toyota Corolla", 2019, 25000, "Müsait")

arac2 = Arac(2, "Honda Civic", 2020, 30000, "Müsait")

arac3 = Arac(3, "Ford Focus", 2018, 20000, "Kiralandı")

arac4 = Arac(4, "Volkswagen Golf", 2017, 18000, "Müsait")

arac5 = Arac(5, "Renault Megane", 2019, 22000, "Kiralandı")

class Musteri:

def \_\_init\_\_(self, adi, soyadi, telefon, kredi\_kart\_bilgileri):

self.adi = adi

self.soyadi = soyadi

self.telefon = telefon

self.kredi\_kart\_bilgileri = kredi\_kart\_bilgileri

# Kullanıcıdan müşteri bilgilerini al

adi = input("Adınız: ")

soyadi = input("Soyadınız: ")

telefon = input("Telefon Numaranız: ")

kredi\_kart\_bilgileri = input("Kredi Kartı Bilgileriniz: ")

musteri = Musteri(adi, soyadi, telefon, kredi\_kart\_bilgileri)

class Kiralama:

def kiralama\_yap(self, musteri, arac):

if arac.kiralama\_durumu == "Müsait":

arac.arac\_durumu\_guncelle("Kiralandı")

print(f"{musteri.adi} {musteri.soyadi}, {arac.model} aracını kiraladı.")

else:

print("Araba şu anda kiralanamaz.")

def kiralama\_iptal\_et(self, arac):

if arac.kiralama\_durumu == "Kiralandı":

arac.arac\_durumu\_guncelle("Müsait")

print(f"{arac.model} aracının kiralama işlemi iptal edildi.")

else:

print("Araba zaten kiralanmamış.")

def kiralama\_bilgisi(self, arac):

print(f"{arac.model} aracının kiralama durumu: {arac.kiralama\_durumu}")

# Örnek kullanım

kiralama = Kiralama()

kiralama.kiralama\_yap(musteri, arac1)

kiralama.kiralama\_bilgisi(arac1)

kiralama.kiralama\_iptal\_et(arac1)

kiralama.kiralama\_bilgisi(arac1)

Arayuz

from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QPushButton, QVBoxLayout, QComboBox, QLineEdit

# Araç sınıfı ve araç verileri burada tanımlanmış olarak varsayılmıştır

app = QApplication([])

window = QWidget()

window.setWindowTitle("Araç Kiralama Sistemi")

label\_adi = QLabel("Adınız:")

input\_adi = QLineEdit()

label\_soyadi = QLabel("Soyadınız:")

input\_soyadi = QLineEdit()

label\_telefon = QLabel("Telefon Numaranız:")

input\_telefon = QLineEdit()

label\_kredi\_kart = QLabel("Kredi Kartı Bilgileriniz:")

input\_kredi\_kart = QLineEdit()

label\_arac\_sec = QLabel("Araç Seçimi:")

combo\_arac\_sec = QComboBox()

button\_kiralama\_yap = QPushButton("Kiralama Yap")

button\_kiralama\_iptal\_et = QPushButton("Kiralama İptal Et")

layout = QVBoxLayout()

layout.addWidget(label\_adi)

layout.addWidget(input\_adi)

layout.addWidget(label\_soyadi)

layout.addWidget(input\_soyadi)

layout.addWidget(label\_telefon)

layout.addWidget(input\_telefon)

layout.addWidget(label\_kredi\_kart)

layout.addWidget(input\_kredi\_kart)

layout.addWidget(label\_arac\_sec)

layout.addWidget(combo\_arac\_sec)

layout.addWidget(button\_kiralama\_yap)

layout.addWidget(button\_kiralama\_iptal\_et)

window.setLayout(layout)

window.show()

app.exec\_()

Изображение выглядит как программное обеспечение, Мультимедийное программное обеспечение, снимок экрана, Значок на компьютере

Автоматически созданное описание

Proje 2

class Doktor:

def \_\_init\_\_(self, isim, uzmanlik\_alani):

self.isim = isim

self.uzmanlik\_alani = uzmanlik\_alani

self.musaitlik\_durumu = True # Başlangıçta doktor müsait kabul edilir

def musait\_mi(self):

return self.musaitlik\_durumu

def randevu\_ayarla(self):

self.musaitlik\_durumu = False

def randevu\_iptal(self):

self.musaitlik\_durumu = True

class Hasta:

def \_\_init\_\_(self, tc, ad\_soyad, dogum\_tarihi, sigorta\_durumu, sigorta\_numarasi=None):

self.tc = tc

self.ad\_soyad = ad\_soyad

self.dogum\_tarihi = dogum\_tarihi

self.sigorta\_durumu = sigorta\_durumu

self.sigorta\_numarasi = sigorta\_numarasi

self.randevu\_gecmisi = []

def randevu\_ekle(self, randevu):

self.randevu\_gecmisi.append(randevu)

class Randevu:

def \_\_init\_\_(self, tarih, doktor, hasta):

self.tarih = tarih

self.doktor = doktor

self.hasta = hasta

def randevu\_al(doktorlar, hasta):

musait\_doktorlar = [doktor for doktor in doktorlar if doktor.musait\_mi()]

if not musait\_doktorlar:

print("Şu anda müsait doktor bulunmamaktadır.")

return

print("Müsait doktorlar:")

for idx, doktor in enumerate(musait\_doktorlar):

print(f"{idx + 1}. {doktor.isim} - {doktor.uzmanlik\_alani}")

try:

secim = int(input("Randevu almak istediğiniz doktorun numarasını girin: ")) - 1

if 0 <= secim < len(musait\_doktorlar):

tarih = input("Randevu tarihi girin (GG/AA/YYYY): ")

yeni\_randevu = Randevu(tarih, musait\_doktorlar[secim], hasta)

musait\_doktorlar[secim].randevu\_ayarla()

hasta.randevu\_ekle(yeni\_randevu)

print(f"Randevunuz {tarih} tarihine alındı.")

else:

print("Geçersiz seçim.")

except ValueError:

print("Lütfen geçerli bir numara girin.")

def randevu\_iptal(hasta):

if not hasta.randevu\_gecmisi:

print("Hiç randevunuz bulunmamaktadır.")

return

for idx, randevu in enumerate(hasta.randevu\_gecmisi):

print(f"{idx + 1}. {randevu.tarih} - {randevu.doktor.isim}")

try:

secim = int(input("İptal etmek istediğiniz randevunun numarasını girin: ")) - 1

if 0 <= secim < len(hasta.randevu\_gecmisi):

iptal\_edilen = hasta.randevu\_gecmisi.pop(secim)

iptal\_edilen.doktor.randevu\_iptal()

print("Randevunuz iptal edildi.")

else:

print("Geçersiz seçim.")

except ValueError:

print("Lütfen geçerli bir numara girin.")

# Örnek kullanım:

doktorlar = [Doktor("Ahmet Yılmaz", "Kardiyoloji"), Doktor("Elif Kaya", "Nöroloji")]

hasta = Hasta("12345678901", "Ayşe Demir", "01/01/1980", True, "1234")

randevu\_al(doktorlar, hasta)

randevu\_iptal(hasta)

arayüz proje 2

import sys

from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QPushButton, QLabel, QComboBox, QLineEdit

class Doktor:

def \_init\_(self, isim, uzmanlik\_alani):

self.isim = isim

self.uzmanlik\_alani = uzmanlik\_alani

self.randevular = []

def randevu\_ayarla(self):

print(f"{self.isim} için yeni bir randevu ayarlandı.")

def randevu\_iptal(self):

print(f"{self.isim} için bir randevu iptal edildi.")

class Hasta:

def \_init\_(self, adi, soyadi, dogum\_tarihi, telefon\_numarasi, sigorta\_numarasi):

self.adi = adi

self.soyadi = soyadi

self.dogum\_tarihi = dogum\_tarihi

self.telefon\_numarasi = telefon\_numarasi

self.sigorta\_numarasi = sigorta\_numarasi

self.randevu\_gecmisi = []

def randevu\_ekle(self, randevu):

self.randevu\_gecmisi.append(randevu)

print(f"{self.adi} {self.soyadi} için yeni bir randevu eklendi: {randevu.tarih}")

class Randevu:

def \_init\_(self, tarih, doktor, hasta):

self.tarih = tarih

self.doktor = doktor

self.hasta = hasta

class RandevuSistemi(QWidget):

def \_init\_(self):

super().\_init\_()

self.doktorlar = [Doktor("Ahmet Yılmaz", "Kardiyoloji"), Doktor("Elif Kaya", "Nöroloji")]

self.initUI()

def initUI(self):

self.layout = QVBoxLayout()

# Hasta bilgileri

self.adiLabel = QLabel("Adı:")

self.layout.addWidget(self.adiLabel)

self.adiLineEdit = QLineEdit()

self.layout.addWidget(self.adiLineEdit)

self.soyadiLabel = QLabel("Soyadı:")

self.layout.addWidget(self.soyadiLabel)

self.soyadiLineEdit = QLineEdit()

self.layout.addWidget(self.soyadiLineEdit)

self.dogumTarihiLabel = QLabel("Doğum Tarihi (GG/AA/YYYY):")

self.layout.addWidget(self.dogumTarihiLabel)

self.dogumTarihiLineEdit = QLineEdit()

self.layout.addWidget(self.dogumTarihiLineEdit)

self.telefonNumarasiLabel = QLabel("Telefon Numarası:")

self.layout.addWidget(self.telefonNumarasiLabel)

self.telefonNumarasiLineEdit = QLineEdit()

self.layout.addWidget(self.telefonNumarasiLineEdit)

self.sigortaNumarasiLabel = QLabel("Sigorta Numarası:")

self.layout.addWidget(self.sigortaNumarasiLabel)

self.sigortaNumarasiLineEdit = QLineEdit()

self.layout.addWidget(self.sigortaNumarasiLineEdit)

# Doktor seçimi

self.doktorSecimLabel = QLabel("Doktor Seçiniz:")

self.layout.addWidget(self.doktorSecimLabel)

self.doktorSecimComboBox = QComboBox()

for doktor in self.doktorlar:

self.doktorSecimComboBox.addItem(doktor.isim + " - " + doktor.uzmanlik\_alani)

self.layout.addWidget(self.doktorSecimComboBox)

# Randevu tarihi

self.tarihLabel = QLabel("Randevu Tarihi (GG/AA/YYYY):")

self.layout.addWidget(self.tarihLabel)

self.tarihLineEdit = QLineEdit()

self.layout.addWidget(self.tarihLineEdit)

# Randevu butonları

self.randevuAlButton = QPushButton("Randevu Al")

self.randevuAlButton.clicked.connect(self.randevuAl)

self.layout.addWidget(self.randevuAlButton)

# Randevu iptal butonu işlevselliği için ekstra bir fonksiyon yazılmalıdır.

#self.randevuIptalButton.clicked.connect(self.randevuIptal)

#self.layout.addWidget(self.randevuIptalButton)

self.setLayout(self.layout)

self.setWindowTitle('Randevu Sistemi')

self.show()

def randevuAl(self):

adi = self.adiLineEdit.text()

soyadi = self.soyadiLineEdit.text()

dogum\_tarihi = self.dogumTarihiLineEdit.text()

telefon\_numarasi = self.telefonNumarasiLineEdit.text()

# Create Randevu object

randevu = Randevu(self.tarihLineEdit.text(), None, None)

# Create Hasta object

hasta = Hasta(adi, soyadi, dogum\_tarihi, telefon\_numarasi, None)

hasta.randevu\_ekle(randevu)

if \_name\_ == '\_main\_':

app = QApplication(sys.argv)

ex = RandevuSistemi()

sys.exit(app.exec\_())

Изображение выглядит как текст, программное обеспечение, компьютер, снимок экрана

Автоматически созданное описание

Proje 3

class Kitap:

def \_\_init\_\_(self, kitap\_id, ad, yazar, yil, kategori):

self.kitap\_id = kitap\_id

self.ad = ad

self.yazar = yazar

self.yil = yil

self.kategori = kategori

self.durum = 'Mevcut' # Başlangıç durumu

def durum\_guncelle(self, yeni\_durum):

self.durum = yeni\_durum

print(f"{self.ad} kitabının durumu '{yeni\_durum}' olarak güncellendi.")

class Uye:

uye\_listesi = {} # Tüm üyeleri tutan sınıf değişkeni

def \_\_init\_\_(self, uye\_id, ad, soyad, uye\_oldugu\_yil):

self.uye\_id = uye\_id

self.ad = ad

self.soyad = soyad

self.uye\_oldugu\_yil = uye\_oldugu\_yil

self.once\_alinan\_kitaplar = []

Uye.uye\_listesi[uye\_id] = self # Üye oluşturulduğunda listeye eklenir

def kitap\_ekle(self, kitap):

self.once\_alinan\_kitaplar.append(kitap)

print(f"{kitap.ad}, {self.ad} {self.soyad} tarafından daha önce ödünç alındı.")

@classmethod

def uye\_bilgisi\_getir(cls, uye\_id):

uye = cls.uye\_listesi.get(uye\_id, None)

if uye:

print(f"Üye ID: {uye.uye\_id}, Ad: {uye.ad}, Soyad: {uye.soyad}, Üyelik Yılı: {uye.uye\_oldugu\_yil}")

for kitap in uye.once\_alinan\_kitaplar:

print(f"Ödünç Alınan Kitap: {kitap.ad}")

else:

print("Bu ID'ye sahip bir üye bulunamadı. Lütfen üye olunuz.")

class Odunc:

def \_\_init\_\_(self):

self.odunc\_listesi = []

def odunc\_al(self, uye, kitap):

if kitap.durum == 'Mevcut':

kitap.durum\_guncelle('Ödünç Alındı')

uye.kitap\_ekle(kitap)

self.odunc\_listesi.append((uye, kitap))

print(f"{kitap.ad} kitabı {uye.ad} {uye.soyad} tarafından ödünç alındı.")

else:

print(f"{kitap.ad} kitabı şu anda mevcut değil.")

def iade\_et(self, uye, kitap):

if (uye, kitap) in self.odunc\_listesi:

kitap.durum\_guncelle('Mevcut')

self.odunc\_listesi.remove((uye, kitap))

print(f"{kitap.ad} kitabı {uye.ad} {uye.soyad} tarafından iade edildi.")

else:

print("Bu ödünç kaydı bulunamadı.")

def odunc\_bilgisi(self):

for uye, kitap in self.odunc\_listesi:

print(f"{uye.ad} {uye.soyad}, {kitap.ad} ({kitap.yazar}) kitabını ödünç aldı.")

Arayüz proje 3

import sys

from PyQt5.QtWidgets import QApplication, QWidget, QPushButton, QVBoxLayout, QLabel, QLineEdit, QMessageBox

class LibraryGUI(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.initUI()

def initUI(self):

self.setWindowTitle('Kütüphane Yönetim Sistemi')

self.setGeometry(100, 100, 400, 300)

layout = QVBoxLayout()

self.uyeIdInput = QLineEdit(self)

self.uyeIdInput.setPlaceholderText('Üye ID')

layout.addWidget(self.uyeIdInput)

self.uyeAdiInput = QLineEdit(self)

self.uyeAdiInput.setPlaceholderText('Üye Adı')

layout.addWidget(self.uyeAdiInput)

self.uyeSoyadiInput = QLineEdit(self)

self.uyeSoyadiInput.setPlaceholderText('Üye Soyadı')

layout.addWidget(self.uyeSoyadiInput)

self.kitapIdInput = QLineEdit(self)

self.kitapIdInput.setPlaceholderText('Kitap ID')

layout.addWidget(self.kitapIdInput)

self.kitapAdiInput = QLineEdit(self)

self.kitapAdiInput.setPlaceholderText('Kitap Adı')

layout.addWidget(self.kitapAdiInput)

uyeEkleBtn = QPushButton('Üye Ekle', self)

uyeEkleBtn.clicked.connect(self.uyeEkle)

layout.addWidget(uyeEkleBtn)

kitapEkleBtn = QPushButton('Kitap Ekle', self)

kitapEkleBtn.clicked.connect(self.kitapEkle)

layout.addWidget(kitapEkleBtn)

oduncAlBtn = QPushButton('Ödünç Al', self)

oduncAlBtn.clicked.connect(self.oduncAl)

layout.addWidget(oduncAlBtn)

iadeEtBtn = QPushButton('Kitabı İade Et', self)

iadeEtBtn.clicked.connect(self.iadeEt)

layout.addWidget(iadeEtBtn)

self.setLayout(layout)

def uyeEkle(self):

# Buraya üye ekleme işlevi kodları gelecek

QMessageBox.information(self, 'Bilgi', 'Üye Eklendi')

def kitapEkle(self):

# Buraya kitap ekleme işlevi kodları gelecek

QMessageBox.information(self, 'Bilgi', 'Kitap Eklendi')

def oduncAl(self):

# Buraya ödünç alma işlevi kodları gelecek

QMessageBox.information(self, 'Bilgi', 'Kitap Ödünç Alındı')

def iadeEt(self):

# Buraya kitap iade işlevi kodları gelecek

QMessageBox.information(self, 'Bilgi', 'Kitap İade Edildi')

def main():

app = QApplication(sys.argv)

ex = LibraryGUI()

ex.show()

sys.exit(app.exec\_())

if \_\_name\_\_ == '\_\_main\_\_':

main()

Изображение выглядит как снимок экрана, программное обеспечение, текст, Мультимедийное программное обеспечение

Автоматически созданное описание

Proje 4

class Etkinlik:

def \_\_init\_\_(self, adi, tarihi, yerleskeni, saati, fiyati):

self.adi = adi

self.tarihi = tarihi

self.yerleskeni = yerleskeni

self.saati = saati

self.fiyati = fiyati

class Katilimci:

def \_\_init\_\_(self, adi, soyadi, dogum\_tarihi):

self.adi = adi

self.soyadi = soyadi

self.dogum\_tarihi = dogum\_tarihi

class Bilet:

def \_\_init\_\_(self, etkinlik, katilimci):

self.etkinlik = etkinlik

self.katilimci = katilimci

def satin\_al(self, kredi\_karti\_bilgileri):

# Burada gerçek bir ödeme işlemi yapılacak.

# Güvenlik nedeniyle bu adımı basitleştiriyoruz.

print(f"{self.katilimci.adi} adına {self.etkinlik.adi} etkinliğine bilet satın alındı.")

class EtkinlikSistemi:

def \_\_init\_\_(self):

self.etkinlikler = []

self.katilimcilar = []

def etkinlik\_ekle(self, etkinlik):

self.etkinlikler.append(etkinlik)

def katilimci\_ekle(self, katilimci):

self.katilimcilar.append(katilimci)

def bilet\_satin\_al(self, etkinlik\_adi, katilimci\_adi, kredi\_karti\_bilgileri):

etkinlik = next((e for e in self.etkinlikler if e.adi == etkinlik\_adi), None)

katilimci = next((k for k in self.katilimcilar if k.adi == katilimci\_adi), None)

if etkinlik and katilimci:

bilet = Bilet(etkinlik, katilimci)

bilet.satin\_al(kredi\_karti\_bilgileri)

else:

print("Etkinlik veya Katılımcı bulunamadı.")

# Sistem oluşturma

sistem = EtkinlikSistemi()

# Etkinlikleri Ekleme

sistem.etkinlik\_ekle(Etkinlik("Konser", "2023-06-15", "Stadyum", "20:00", 150))

sistem.etkinlik\_ekle(Etkinlik("Tiyatro", "2023-06-20", "Kültür Merkezi", "19:00", 100))

# Katılımcıları Ekleme

sistem.katilimci\_ekle(Katilimci("Ali", "Veli", "1990-01-01"))

sistem.katilimci\_ekle(Katilimci("Ayşe", "Yılmaz", "1985-05-05"))

# Bilet Satın Alma

sistem.bilet\_satin\_al("Konser", "Ali", {"kart\_numarasi": "1234-5678-9012-3456", "son\_kullanma\_tarihi": "06/25", "cvv": "123"})

proje 4 arayüz

import sys

from PyQt5.QtWidgets import QApplication, QWidget, QPushButton, QVBoxLayout, QLineEdit, QFormLayout, QMessageBox, QComboBox, QRadioButton

class EtkinlikSistemiGUI(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.initUI()

def initUI(self):

self.setWindowTitle('Etkinlik Sistemi')

self.setGeometry(100, 100, 280, 220)

# Form Layout Oluştur

layout = QFormLayout()

# Kullanıcı Giriş Alanları

self.etkinlikAdiInput = QLineEdit(self)

self.etkinlikTurleri = QComboBox(self)

self.etkinlikTurleri.addItems(['Spor', 'Müzik', 'Tiyatro'])

self.katilimciAdiInput = QLineEdit(self)

self.krediKartiBilgileriInput = QLineEdit(self)

# Cinsiyet Seçimi

self.cinsiyetSecimi = QRadioButton('Erkek', self)

self.cinsiyetSecimi.setChecked(True)

self.cinsiyetSecimi2 = QRadioButton('Kadın', self)

# Giriş alanlarını layout'a ekle

layout.addRow('Etkinlik Adı:', self.etkinlikAdiInput)

layout.addRow('Etkinlik Türü:', self.etkinlikTurleri)

layout.addRow('Katılımcı Adı:', self.katilimciAdiInput)

layout.addRow('Kredi Kartı Bilgileri:', self.krediKartiBilgileriInput)

layout.addRow('Cinsiyet:', self.cinsiyetSecimi)

layout.addRow('', self.cinsiyetSecimi2)

# Bilet Satın Al Butonu

self.biletSatinAlBtn = QPushButton('Bilet Satın Al', self)

self.biletSatinAlBtn.clicked.connect(self.biletSatinAl)

layout.addWidget(self.biletSatinAlBtn)

self.setLayout(layout)

def biletSatinAl(self):

etkinlikAdi = self.etkinlikAdiInput.text()

secilenEtkinlikTuru = self.etkinlikTurleri.currentText()

katilimciAdi = self.katilimciAdiInput.text()

krediKartiBilgileri = self.krediKartiBilgileriInput.text()

cinsiyet = 'Erkek' if self.cinsiyetSecimi.isChecked() else 'Kadın'

# Basitçe bilgileri bir mesaj kutusunda göster

QMessageBox.information(self, 'Bilet Satın Alındı', f"Etkinlik Adı: {etkinlikAdi}\nEtkinlik Türü: {secilenEtkinlikTuru}\nKatılımcı: {katilimciAdi}\nKredi Kartı Bilgileri: {krediKartiBilgileri}\nCinsiyet: {cinsiyet}", QMessageBox.Ok)

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

ex = EtkinlikSistemiGUI()

ex.show()

sys.exit(app.exec\_())

Изображение выглядит как программное обеспечение, Мультимедийное программное обеспечение, текст, Значок на компьютере

Автоматически созданное описание

**Proje 5: Online Eğitim Platformu**

from flask import Flask, render\_template, request, redirect, url\_for

class Kurs:

def \_\_init\_\_(self, id, adi, egitmen\_adi, sure, fiyat, tipi):

self.id = id

self.adi = adi

self.egitmen\_adi = egitmen\_adi

self.sure = sure

self.fiyat = fiyat

self.tipi = tipi # Online veya Offline

self.kayitli\_ogrenciler = []

def ogrenci\_kaydet(self, ogrenci):

self.kayitli\_ogrenciler.append(ogrenci)

class Egitmen:

def \_\_init\_\_(self, adi, soyadi, uzmanlik\_alani, egitim\_aldigi\_kurum):

self.adi = adi

self.soyadi = soyadi

self.uzmanlik\_alani = uzmanlik\_alani

self.egitim\_aldigi\_kurum = egitim\_aldigi\_kurum

class Ogrenci:

def \_\_init\_\_(self, adi, soyadi, yas, e\_posta, egitim\_aldigi\_yer, sinifi):

self.adi = adi

self.soyadi = soyadi

self.yas = yas

self.e\_posta = e\_posta

self.egitim\_aldigi\_yer = egitim\_aldigi\_yer

self.sinifi = sinifi

app = Flask(\_\_name\_\_)

kurslar = [

Kurs(id=1, adi="Python Programlama", egitmen\_adi="Ahmet Yılmaz", sure="30 Saat", fiyat=500, tipi="Online"),

]

egitmenler = []

ogrenciler = []

@app.route('/')

def ana\_sayfa():

return render\_template('index.html', kurslar=kurslar)

@app.route('/kaydol/<int:kurs\_id>', methods=['GET', 'POST'])

def kaydol(kurs\_id):

if request.method == 'POST':

adi = request.form['adi']

soyadi = request.form['soyadi']

yas = request.form['yas']

e\_posta = request.form['e\_posta']

egitim\_aldigi\_yer = request.form['egitim\_aldigi\_yer']

sinifi = request.form['sinifi']

yeni\_ogrenci = Ogrenci(adi=adi, soyadi=soyadi, yas=yas, e\_posta=e\_posta, egitim\_aldigi\_yer=egitim\_aldigi\_yer, sinifi=sinifi)

ogrenciler.append(yeni\_ogrenci)

for kurs in kurslar:

if kurs.id == kurs\_id:

kurs.ogrenci\_kaydet(yeni\_ogrenci)

break

return redirect(url\_for('ana\_sayfa'))

return render\_template('kaydol.html', kurs\_id=kurs\_id)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Online Eğitim Platformu</title>

</head>

<body>

<h1>Kurs Listesi</h1>

<ul>

{% for kurs in kurslar %}

<li>{{ kurs.adi }} - <a href="{{ url\_for('kaydol', kurs\_id=kurs.id) }}">Kaydol</a></li>

{% endfor %}

</ul>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Kaydol</title>

</head>

<body>

<h1>Kursa Kaydol</h1>

<form method="post">

Adı: <input type="text" name="adi"><br>

Soyadı: <input type="text" name="soyadi"><br>

Yaş: <input type="text" name="yas"><br>

E-posta: <input type="email" name="e\_posta"><br>

Eğitim Aldığı Yer: <input type="text" name="egitim\_aldigi\_yer"><br>

Sınıfı: <input type="text" name="sinifi"><br>

<input type="submit" value="Kaydol">

</form>

</body>

</html>

Arayüz proje 5.

import sys

from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget, QVBoxLayout, QHBoxLayout, QLabel, QLineEdit, QPushButton, QListWidget, QMessageBox

import sqlite3

class OnlineEgitimPlatformu(QMainWindow):

def \_init\_(self):

super().\_init\_()

self.setWindowTitle("Online Eğitim Platformu")

self.setGeometry(100, 100, 350, 200)

self.central\_widget = QWidget()

self.setCentralWidget(self.central\_widget)

self.layout = QVBoxLayout()

self.central\_widget.setLayout(self.layout)

self.kurs\_listesi = QListWidget()

self.layout.addWidget(self.kurs\_listesi)

self.kurslari\_yukle()

self.kaydol\_button = QPushButton("Kursa Kaydol")

self.kaydol\_button.clicked.connect(self.kursa\_kaydol)

self.layout.addWidget(self.kaydol\_button)

self.icerigi\_goruntule\_button = QPushButton("İçeriği Görüntüle")

self.icerigi\_goruntule\_button.clicked.connect(self.icerigi\_goruntule)

self.layout.addWidget(self.icerigi\_goruntule\_button)

self.yeni\_kurs\_olustur\_button = QPushButton("Yeni Kurs Oluştur")

self.yeni\_kurs\_olustur\_button.clicked.connect(self.yeni\_kurs\_olustur\_dialog\_ac)

self.layout.addWidget(self.yeni\_kurs\_olustur\_button)

self.yeni\_egitmen\_ekle\_button = QPushButton("Yeni Eğitmen Ekle")

self.yeni\_egitmen\_ekle\_button.clicked.connect(self.yeni\_egitmen\_ekle\_dialog\_ac)

self.layout.addWidget(self.yeni\_egitmen\_ekle\_button)

self.ogrenci\_kaydi\_yap\_button = QPushButton("Öğrenci Kaydı Yap")

self.ogrenci\_kaydi\_yap\_button.clicked.connect(self.ogrenci\_kaydi\_dialog\_ac)

self.layout.addWidget(self.ogrenci\_kaydi\_yap\_button)

def kurslari\_yukle(self):

self.kurs\_listesi.clear()

baglanti = sqlite3.connect("courses.db")

imlec = baglanti.cursor()

imlec.execute("SELECT \* FROM courses")

kurslar = imlec.fetchall()

for kurs in kurslar:

self.kurs\_listesi.addItem(kurs[1]) # Kurs adının ikinci sütunda olduğunu varsayalım

baglanti.close()

def kursa\_kaydol(self):

secili\_kurs = self.kurs\_listesi.currentItem()

if secili\_kurs:

kurs\_adi = secili\_kurs.text()

QMessageBox.information(self, "Kayıt Başarılı", f"{kurs\_adi} kursuna başarıyla kaydoldunuz!")

else:

QMessageBox.warning(self, "Hata", "Lütfen bir kurs seçin!")

def icerigi\_goruntule(self):

secili\_kurs = self.kurs\_listesi.currentItem()

if secili\_kurs:

kurs\_adi = secili\_kurs.text()

QMessageBox.information(self, "İçerik Görüntüleme", f"{kurs\_adi} kursunun içeriği yükleniyor...")

# Burada kurs içeriğinin yüklenmesi veya gösterilmesi işlemi yapılabilir.

else:

QMessageBox.warning(self, "Hata", "Lütfen bir kurs seçin!")

def yeni\_kurs\_olustur\_dialog\_ac(self):

dialog = KursOlusturDialog()

dialog.exec\_()

self.kurslari\_yukle()

def yeni\_egitmen\_ekle\_dialog\_ac(self):

dialog = EgitmenOlusturDialog()

dialog.exec\_()

def ogrenci\_kaydi\_dialog\_ac(self):

dialog = OgrenciKaydiDialog()

dialog.exec\_()

class KursOlusturDialog(QWidget):

def \_init\_(self):

super().\_init\_()

self.setWindowTitle("Yeni Kurs Oluştur")

self.setGeometry(200, 200, 400, 200)

self.layout = QVBoxLayout()

self.setLayout(self.layout)

self.kurs\_adi\_label = QLabel("Kurs Adı:")

self.kurs\_adi\_input = QLineEdit()

self.layout.addWidget(self.kurs\_adi\_label)

self.layout.addWidget(self.kurs\_adi\_input)

self.egitmen\_label = QLabel("Eğitmen:")

self.egitmen\_input = QLineEdit()

self.layout.addWidget(self.egitmen\_label)

self.layout.addWidget(self.egitmen\_input)

self.icerik\_label = QLabel("İçerik:")

self.icerik\_input = QLineEdit()

self.layout.addWidget(self.icerik\_label)

self.layout.addWidget(self.icerik\_input)

self.olustur\_button = QPushButton("Oluştur")

self.layout.addWidget(self.olustur\_button)

self.olustur\_button.clicked.connect(self.kurs\_olustur)

def kurs\_olustur(self):

baglanti = sqlite3.connect("courses.db")

imlec = baglanti.cursor()

kurs\_adi = self.kurs\_adi\_input.text()

egitmen = self.egitmen\_input.text()

icerik = self.icerik\_input.text()

imlec.execute("INSERT INTO courses (course\_name, trainer, content) VALUES (?, ?, ?)", (kurs\_adi, egitmen, icerik))

baglanti.commit()

baglanti.close()

self.close()

class EgitmenOlusturDialog(QWidget):

def \_init\_(self):

super().\_init\_()

self.setWindowTitle("Yeni Eğitmen Ekle")

self.setGeometry(200, 200, 400, 200)

self.layout = QVBoxLayout()

self.setLayout(self.layout)

self.egitmen\_adi\_label = QLabel("Eğitmen Adı:")

self.egitmen\_adi\_input = QLineEdit()

self.layout.addWidget(self.egitmen\_adi\_label)

self.layout.addWidget(self.egitmen\_adi\_input)

self.uzmanlik\_alani\_label = QLabel("Uzmanlık Alanı:")

self.uzmanlik\_alani\_input = QLineEdit()

self.layout.addWidget(self.uzmanlik\_alani\_label)

self.layout.addWidget(self.uzmanlik\_alani\_input)

self.ekle\_button = QPushButton("Ekle")

self.layout.addWidget(self.ekle\_button)

self.ekle\_button.clicked.connect(self.egitmen\_ekle)

def egitmen\_ekle(self):

baglanti = sqlite3.connect("trainers.db")

imlec = baglanti.cursor()

egitmen\_adi = self.egitmen\_adi\_input.text()

uzmanlik\_alani = self.uzmanlik\_alani\_input.text()

imlec.execute("INSERT INTO trainers (name, expertise) VALUES (?, ?)", (egitmen\_adi, uzmanlik\_alani))

baglanti.commit()

baglanti.close()

self.close()

class OgrenciKaydiDialog(QWidget):

def \_init\_(self):

super().\_init\_()

self.setWindowTitle("Öğrenci Kaydı Yap")

self.setGeometry(200, 200, 400, 200)

self.layout = QVBoxLayout()

self.setLayout(self.layout)

self.ogrenci\_adi\_label = QLabel("Öğrenci Adı:")

self.ogrenci\_adi\_input = QLineEdit()

self.layout.addWidget(self.ogrenci\_adi\_label)

self.layout.addWidget(self.ogrenci\_adi\_input)

self.eposta\_label = QLabel("E-posta:")

self.eposta\_input = QLineEdit()

self.layout.addWidget(self.eposta\_label)

self.layout.addWidget(self.eposta\_input)

self.kaydol\_button = QPushButton("Kaydol")

self.layout.addWidget(self.kaydol\_button)

self.kaydol\_button.clicked.connect(self.ogrenci\_kaydi)

def ogrenci\_kaydi(self):

baglanti = sqlite3.connect("students.db")

imlec = baglanti.cursor()

ogrenci\_adi = self.ogrenci\_adi\_input.text()

eposta = self.eposta\_input.text()

imlec.execute("INSERT INTO students (name, email) VALUES (?, ?)", (ogrenci\_adi, eposta))

baglanti.commit()

baglanti.close()

self.close()

if \_name\_ == "\_main\_":

app = QApplication(sys.argv)

# Veritabanı oluşturma ve tabloyu oluşturma

baglanti\_kurslar = sqlite3.connect("courses.db")

imlec\_kurslar = baglanti\_kurslar.cursor()

imlec\_kurslar.execute("CREATE TABLE IF NOT EXISTS courses (id INTEGER PRIMARY KEY, course\_name TEXT, trainer TEXT, content TEXT)")

baglanti\_kurslar.close()

baglanti\_egitmenler = sqlite3.connect("trainers.db")

imlec\_egitmenler = baglanti\_egitmenler.cursor()

imlec\_egitmenler.execute("CREATE TABLE IF NOT EXISTS trainers (id INTEGER PRIMARY KEY, name TEXT, expertise TEXT)")

baglanti\_egitmenler.close()

baglanti\_ogrenciler = sqlite3.connect("students.db")

imlec\_ogrenciler = baglanti\_ogrenciler.cursor()

imlec\_ogrenciler.execute("CREATE TABLE IF NOT EXISTS students (id INTEGER PRIMARY KEY, name TEXT, email TEXT)")

baglanti\_ogrenciler.close()

ana\_pencere = OnlineEgitimPlatformu()

ana\_pencere.show()

sys.exit(app.exec\_())

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Автоматически созданное описание

Veriyle bağli arayüzler

Proje 1

from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QPushButton, QVBoxLayout, QComboBox, QLineEdit

class Arac:

def \_\_init\_\_(self, id, model, yil, km, durum):

self.id = id

self.model = model

self.yil = yil

self.km = km

self.durum = durum

def \_\_str\_\_(self):

return f"{self.model} ({self.yil}) - {self.durum}"

# Araç verileri

arac\_listesi = [

Arac(1, "Toyota Corolla", 2019, 25000, "Müsait"),

Arac(2, "Honda Civic", 2020, 30000, "Müsait"),

Arac(3, "Ford Focus", 2018, 20000, "Kiralandı"),

Arac(4, "Volkswagen Golf", 2017, 18000, "Müsait"),

Arac(5, "Renault Megane", 2019, 22000, "Kiralandı")

]

app = QApplication([])

window = QWidget()

window.setWindowTitle("Araç Kiralama Sistemi")

label\_adi = QLabel("Adınız:")

input\_adi = QLineEdit()

label\_soyadi = QLabel("Soyadınız:")

input\_soyadi = QLineEdit()

label\_telefon = QLabel("Telefon Numaranız:")

input\_telefon = QLineEdit()

label\_kredi\_kart = QLabel("Kredi Kartı Bilgileriniz:")

input\_kredi\_kart = QLineEdit()

label\_arac\_sec = QLabel("Araç Seçimi:")

combo\_arac\_sec = QComboBox()

# Araç listesini combobox'a ekleyin

for arac in arac\_listesi:

if arac.durum == "Müsait":

combo\_arac\_sec.addItem(str(arac), arac)

button\_kiralama\_yap = QPushButton("Kiralama Yap")

button\_kiralama\_iptal\_et = QPushButton("Kiralama İptal Et")

layout = QVBoxLayout()

layout.addWidget(label\_adi)

layout.addWidget(input\_adi)

layout.addWidget(label\_soyadi)

layout.addWidget(input\_soyadi)

layout.addWidget(label\_telefon)

layout.addWidget(input\_telefon)

layout.addWidget(label\_kredi\_kart)

layout.addWidget(input\_kredi\_kart)

layout.addWidget(label\_arac\_sec)

layout.addWidget(combo\_arac\_sec)

layout.addWidget(button\_kiralama\_yap)

layout.addWidget(button\_kiralama\_iptal\_et)

window.setLayout(layout)

# Fonksiyonlar ve Sinyal-Slot bağlantıları burada tanımlanabilir.

# Örneğin:

# def kiralamaYap():

# secili\_arac = combo\_arac\_sec.currentData()

# # Kiralama işlemleri burada gerçekleştirilebilir.

#

# button\_kiralama\_yap.clicked.connect(kiralamaYap)

window.show()

app.exec\_()

proje 3

Proje 4

import sys

from PyQt5.QtWidgets import QApplication, QWidget, QPushButton, QVBoxLayout, QLineEdit, QFormLayout, QMessageBox, QComboBox, QRadioButton, QLabel

class EtkinlikSistemiGUI(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.initUI()

def initUI(self):

self.setWindowTitle('Etkinlik Sistemi')

self.setGeometry(100, 100, 280, 320)

# Form Layout Oluştur

layout = QFormLayout()

# Kullanıcı Giriş Alanları

self.etkinlikAdiInput = QLineEdit(self)

self.etkinlikTurleri = QComboBox(self)

self.etkinlikTurleri.addItems(['Spor', 'Müzik', 'Tiyatro'])

self.katilimciAdiInput = QLineEdit(self)

self.krediKartiBilgileriInput = QLineEdit(self)

self.etkinlikTarihiInput = QLineEdit(self)

self.etkinlikSaatiInput = QLineEdit(self)

self.katilimUcretiInput = QLineEdit(self)

self.etkinlikYeriInput = QLineEdit(self)

# Cinsiyet Seçimi

self.cinsiyetSecimi = QRadioButton('Erkek', self)

self.cinsiyetSecimi.setChecked(True)

self.cinsiyetSecimi2 = QRadioButton('Kadın', self)

# Giriş alanlarını layout'a ekle

layout.addRow('Etkinlik Adı:', self.etkinlikAdiInput)

layout.addRow('Etkinlik Türü:', self.etkinlikTurleri)

layout.addRow('Katılımcı Adı:', self.katilimciAdiInput)

layout.addRow('Kredi Kartı Bilgileri:', self.krediKartiBilgileriInput)

layout.addRow('Etkinlik Tarihi:', self.etkinlikTarihiInput)

layout.addRow('Etkinlik Saati:', self.etkinlikSaatiInput)

layout.addRow('Katılım Ücreti:', self.katilimUcretiInput)

layout.addRow('Etkinlik Yeri:', self.etkinlikYeriInput)

layout.addRow('Cinsiyet:', self.cinsiyetSecimi)

layout.addRow('', self.cinsiyetSecimi2)

# Bilet Satın Al Butonu

self.biletSatinAlBtn = QPushButton('Bilet Satın Al', self)

self.biletSatinAlBtn.clicked.connect(self.biletSatinAl)

layout.addWidget(self.biletSatinAlBtn)

self.setLayout(layout)

def biletSatinAl(self):

etkinlikAdi = self.etkinlikAdiInput.text()

secilenEtkinlikTuru = self.etkinlikTurleri.currentText()

katilimciAdi = self.katilimciAdiInput.text()

krediKartiBilgileri = self.krediKartiBilgileriInput.text()

etkinlikTarihi = self.etkinlikTarihiInput.text()

etkinlikSaati = self.etkinlikSaatiInput.text()

katilimUcreti = self.katilimUcretiInput.text()

etkinlikYeri = self.etkinlikYeriInput.text()

cinsiyet = 'Erkek' if self.cinsiyetSecimi.isChecked() else 'Kadın'

# Basitçe bilgileri bir mesaj kutusunda göster

QMessageBox.information(self, 'Bilet Satın Alındı', f"Etkinlik Adı: {etkinlikAdi}\nEtkinlik Türü: {secilenEtkinlikTuru}\nKatılımcı: {katilimciAdi}\nKredi Kartı Bilgileri: {krediKartiBilgileri}\nEtkinlik Tarihi: {etkinlikTarihi}\nEtkinlik Saati: {etkinlikSaati}\nKatılım Ücreti: {katilimUcreti}\nEtkinlik Yeri: {etkinlikYeri}\nCinsiyet: {cinsiyet}", QMessageBox.Ok)

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

ex = EtkinlikSistemiGUI()

ex.show()

sys.exit(app.exec\_())

