**Proje 9**

import sys

import sqlite3

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

class SeyahatPlanlama(QWidget):

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

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

self.setWindowTitle("Seyahat Planlama Uygulaması")

self.arayuz\_olustur()

self.veritabani\_baglantisi\_olustur()

def arayuz\_olustur(self):

self.duzen = QVBoxLayout()

# Travel (Seyahat) Section

self.seyahat\_rota\_label = QLabel("Seyahat Rotası:")

self.seyahat\_rota\_combobox = QComboBox()

self.seyahat\_rota\_combobox.addItems(["İstanbul - Ankara", "İzmir - Antalya", "Bodrum - Kapadokya", "Bursa - Trabzon", "Eskişehir - Diyarbakır"])

self.duzen.addWidget(self.seyahat\_rota\_label)

self.duzen.addWidget(self.seyahat\_rota\_combobox)

# Accommodation (Konaklama) Section

self.konaklama\_label = QLabel("Konaklama Tesis Adı:")

self.konaklama\_input = QLineEdit()

self.duzen.addWidget(self.konaklama\_label)

self.duzen.addWidget(self.konaklama\_input)

self.fiyat\_label = QLabel("Fiyat:")

self.fiyat\_input = QLineEdit()

self.duzen.addWidget(self.fiyat\_label)

self.duzen.addWidget(self.fiyat\_input)

self.konaklama\_sec\_button = QPushButton("Konaklama Seç")

self.konaklama\_sec\_button.clicked.connect(self.konaklama\_sec)

self.duzen.addWidget(self.konaklama\_sec\_button)

# Route (Rota) Section

self.rota\_detay\_label = QLabel("Seyahat Rota Detayları:")

self.rota\_detay\_input = QTextEdit()

self.duzen.addWidget(self.rota\_detay\_label)

self.duzen.addWidget(self.rota\_detay\_input)

self.rota\_ekle\_button = QPushButton("Rota Ekle")

self.rota\_ekle\_button.clicked.connect(self.rota\_ekle)

self.duzen.addWidget(self.rota\_ekle\_button)

# Plan List Section

self.plan\_listesi\_label = QLabel("Seyahat Planı:")

self.plan\_listesi = QListWidget()

self.duzen.addWidget(self.plan\_listesi\_label)

self.duzen.addWidget(self.plan\_listesi)

self.setLayout(self.duzen)

def veritabani\_baglantisi\_olustur(self):

self.veritabani\_baglantisi = sqlite3.connect('seyahat\_planlama.db')

self.cursor = self.veritabani\_baglantisi.cursor()

self.cursor.execute('''CREATE TABLE IF NOT EXISTS Rotalar

(id INTEGER PRIMARY KEY,

rota TEXT,

konaklama TEXT,

fiyat INTEGER)''')

self.veritabani\_baglantisi.commit()

def rota\_ekle(self):

rota = self.seyahat\_rota\_combobox.currentText()

konaklama = self.konaklama\_input.text().strip()

fiyat = self.fiyat\_input.text().strip()

rota\_detay = self.rota\_detay\_input.toPlainText().strip()

if rota and konaklama and fiyat and rota\_detay:

self.cursor.execute("INSERT INTO Rotalar (rota, konaklama, fiyat) VALUES (?, ?, ?)", (rota, konaklama, fiyat))

self.veritabani\_baglantisi.commit()

self.plan\_listesini\_yukle()

QMessageBox.information(self, "Başarılı", "Rota başarıyla eklendi!")

else:

QMessageBox.warning(self, "Uyarı", "Lütfen rota, konaklama, fiyat ve rota detay bilgilerini girin.")

def konaklama\_sec(self):

selected\_item = self.plan\_listesi.currentItem()

if selected\_item:

konaklama\_bilgisi = selected\_item.text().split(' - ')[1]

self.konaklama\_input.setText(konaklama\_bilgisi)

else:

QMessageBox.warning(self, "Uyarı", "Lütfen bir konaklama seçeneği seçin.")

def plan\_listesini\_yukle(self):

self.plan\_listesi.clear()

self.cursor.execute("SELECT rota, konaklama, fiyat FROM Rotalar")

rotalar = self.cursor.fetchall()

for rota in rotalar:

self.plan\_listesi.addItem(f"Rota: {rota[0]} - Konaklama: {rota[1]} - Fiyat: {rota[2]}")

def closeEvent(self, event):

self.veritabani\_baglantisi.close()

event.accept()

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

uygulama = QApplication(sys.argv)

pencere = SeyahatPlanlama()

pencere.show()

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

**Proje 10**

import sys

import sqlite3

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

class CRM(QWidget):

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

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

self.setWindowTitle("Müşteri İlişkileri Yönetimi (CRM)")

self.arayuz\_olustur()

self.veritabani\_baglantisi\_olustur()

def arayuz\_olustur(self):

self.duzen = QVBoxLayout()

# Customer (Müşteri) Section

self.musteri\_label = QLabel("Müşteri Adı:")

self.musteri\_input = QLineEdit()

self.duzen.addWidget(self.musteri\_label)

self.duzen.addWidget(self.musteri\_input)

self.iletisim\_label = QLabel("İletişim Bilgileri:")

self.iletisim\_input = QLineEdit()

self.duzen.addWidget(self.iletisim\_label)

self.duzen.addWidget(self.iletisim\_input)

self.musteri\_ekle\_button = QPushButton("Müşteri Ekle")

self.musteri\_ekle\_button.clicked.connect(self.musteri\_ekle)

self.duzen.addWidget(self.musteri\_ekle\_button)

# Sale (Satış) Section

self.satis\_numarasi\_label = QLabel("Satış Numarası:")

self.satis\_numarasi\_input = QLineEdit()

self.duzen.addWidget(self.satis\_numarasi\_label)

self.duzen.addWidget(self.satis\_numarasi\_input)

self.satilan\_urunler\_label = QLabel("Satılan Ürünler:")

self.satilan\_urunler\_input = QLineEdit()

self.duzen.addWidget(self.satilan\_urunler\_label)

self.duzen.addWidget(self.satilan\_urunler\_input)

self.satis\_ekle\_button = QPushButton("Satış Ekle")

self.satis\_ekle\_button.clicked.connect(self.satis\_ekle)

self.duzen.addWidget(self.satis\_ekle\_button)

# Support (Destek) Section

self.destek\_talebi\_label = QLabel("Destek Talebi Detayları:")

self.destek\_talebi\_input = QTextEdit()

self.duzen.addWidget(self.destek\_talebi\_label)

self.duzen.addWidget(self.destek\_talebi\_input)

self.destek\_talebi\_ekle\_button = QPushButton("Destek Talebi Oluştur")

self.destek\_talebi\_ekle\_button.clicked.connect(self.destek\_talebi\_olustur)

self.duzen.addWidget(self.destek\_talebi\_ekle\_button)

self.setLayout(self.duzen)

def veritabani\_baglantisi\_olustur(self):

self.veritabani\_baglantisi = sqlite3.connect('crm.db')

self.cursor = self.veritabani\_baglantisi.cursor()

self.cursor.execute('''CREATE TABLE IF NOT EXISTS Musteriler

(id INTEGER PRIMARY KEY,

ad TEXT,

iletisim TEXT)''')

self.cursor.execute('''CREATE TABLE IF NOT EXISTS Satislar

(id INTEGER PRIMARY KEY,

satis\_numarasi INTEGER,

musteri\_id INTEGER,

urunler TEXT,

FOREIGN KEY(musteri\_id) REFERENCES Musteriler(id))''')

self.cursor.execute('''CREATE TABLE IF NOT EXISTS DestekTalepleri

(id INTEGER PRIMARY KEY,

musteri\_id INTEGER,

detaylar TEXT,

FOREIGN KEY(musteri\_id) REFERENCES Musteriler(id))''')

self.veritabani\_baglantisi.commit()

def musteri\_ekle(self):

musteri\_ad = self.musteri\_input.text().strip()

iletisim\_bilgisi = self.iletisim\_input.text().strip()

if musteri\_ad and iletisim\_bilgisi:

self.cursor.execute("INSERT INTO Musteriler (ad, iletisim) VALUES (?, ?)", (musteri\_ad, iletisim\_bilgisi))

self.veritabani\_baglantisi.commit()

QMessageBox.information(self, "Başarılı", "Müşteri başarıyla eklendi!")

else:

QMessageBox.warning(self, "Uyarı", "Lütfen müşteri adı ve iletişim bilgilerini girin.")

def satis\_ekle(self):

satis\_numarasi = self.satis\_numarasi\_input.text().strip()

musteri\_id = self.musteri\_input.text().strip()

urunler = self.satilan\_urunler\_input.text().strip()

if satis\_numarasi and musteri\_id and urunler:

self.cursor.execute("INSERT INTO Satislar (satis\_numarasi, musteri\_id, urunler) VALUES (?, ?, ?)", (satis\_numarasi, musteri\_id, urunler))

self.veritabani\_baglantisi.commit()

QMessageBox.information(self, "Başarılı", "Satış başarıyla eklendi!")

else:

QMessageBox.warning(self, "Uyarı", "Lütfen satış numarası, müşteri ID ve satılan ürünleri girin.")

def destek\_talebi\_olustur(self):

musteri\_id = self.musteri\_input.text().strip()

detaylar = self.destek\_talebi\_input.toPlainText().strip()

if musteri\_id and detaylar:

self.cursor.execute("INSERT INTO DestekTalepleri (musteri\_id, detaylar) VALUES (?, ?)", (musteri\_id, detaylar))

self.veritabani\_baglantisi.commit()

QMessageBox.information(self, "Başarılı", "Destek talebi başarıyla oluşturuldu!")

else:

QMessageBox.warning(self, "Uyarı", "Lütfen müşteri ID ve destek talebi detaylarını girin.")

def closeEvent(self, event):

self.veritabani\_baglantisi.close()

event.accept()

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

app = QApplication(sys.argv)

window = CRM()

window.show()

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