****

**İsim Soyisim: Salih Karakaya**

**Öğrenci No: 191816775**

1. **Ödeve ait isterler ve bu isterlerin açıklanması**

Bu ödev kapsamında, mobil uygulama geliştirme sürecinde kullanıcı arayüzü tasarımı, veri saklama yöntemleri ve ekranlar arası geçiş işlemlerinin nasıl gerçekleştirileceği ele alınmaktadır. Ödevin temel amacı, uygulamanın her iki ekranında da kullanıcı deneyimini artıracak, okunabilir ve sürdürülebilir bir kod yapısı oluşturmak, aynı zamanda tasarım ve isimlendirme konularında titizlik göstermektir. Aşağıda, gereksinimlerin detaylı açıklaması yer almaktadır:

* **Ana Ekranın Tasarımı ve Logo Kullanımı:**Uygulamanın ana ekranında, sol üst köşede Gazi Üniversitesi logosunun yer alması öngörülmektedir. Bu gereksinim, uygulamanın hem görsel kimliğini güçlendirmesi hem de ilgili kurumla olan bağlantıyı vurgulaması açısından önem taşır. Tasarım sürecinde logonun doğru konumlandırılması ve boyutlandırılması, kullanıcı arayüzünün profesyonel görünmesine katkı sağlar.
* **Kullanıcı Bilgilerinin Alınması:**  
  İlk ekranda kullanıcıdan “Kullanıcı Adı, Konum ve Yaş” bilgileri alınacaktır. Bu bilgiler, kullanıcı dostu EditText bileşenleri ile toplanarak, veri giriş sürecinin basit ve anlaşılır olması sağlanır. Gerekli alanların etiketlendirilmesi, kullanıcı hatalarını en aza indirger.
* **Verilerin SharedPreferences’a Kaydedilmesi ve Ekranlar Arası Geçiş:**  
  Kullanıcının girdiği bilgiler, bir Button aracılığıyla SharedPreferences’a kaydedilecektir. Bu yöntem sayesinde veriler cihaz üzerinde kalıcı olarak saklanır; uygulama kapatılıp açıldığında dahi kullanıcı bilgilerine erişim sağlanabilir. Aynı buton, kaydetme işleminin ardından kullanıcıyı ikinci ekrana yönlendirerek, veri doğrulama ve görüntüleme işlemlerine geçişi kolaylaştırır.
* **İkinci Ekranda Bilgilerin Görüntülenmesi:**  
  İkinci ekranda, SharedPreferences’da saklanan kullanıcı bilgileri, her biri ayrı bir TextView kullanılarak ekranda gösterilecektir. Bu sayede kullanıcı, daha önce girdiği bilgilere kolayca erişebilir ve doğruluğunu kontrol edebilir.
* **Veri Silme İşlemi ve Kullanıcı Geri Bildirimi:**  
  İkinci ekranda, kullanıcıya kaydedilen verilerden istediğini silebilme imkânı tanınacaktır. Bu işlem için bir EditText aracılığıyla silmek istenen verinin anahtarı girilecektir (örneğin: “username”, “location” veya “age”). Girilen anahtar geçerli ise ilgili veri silinecek, aksi durumda ise kullanıcıya yanlış giriş yaptığını belirten bir Toast mesajı gösterilecektir. Bu durum, kullanıcı deneyiminin iyileştirilmesi ve hata yönetiminin sağlanması açısından önemlidir.
* **Ekranlar Arası Geçişin Sağlanması:**  
  İkinci ekranda, kullanıcıya ilk ekrana dönüş yapabilmesi için ek bir Button sunulacaktır. Bu buton, uygulama içindeki navigasyonun kullanıcı dostu bir şekilde gerçekleştirilmesine yardımcı olur.
* **Tasarım Standartları ve İsimlendirme Kuralları:**  
  Uygulama geliştirilirken, kullanıcı arayüzünde kullanılacak font, renk ve genel tasarım ögelerinin uyumlu ve estetik olması beklenmektedir. Ayrıca, sınıf ve değişken isimlendirmelerinde açıklık ve tutarlılık ön planda tutulmalıdır. Bu sayede, kodun okunabilirliği ve sürdürülebilirliği sağlanırken, ekip çalışması ve bakım süreçlerinde de kolaylık elde edilir.

Bu gereksinimler, uygulamanın hem teknik hem de görsel açıdan kullanıcı odaklı bir deneyim sunmasını amaçlamaktadır. Ödev sürecinde, bu isterler doğrultusunda oluşturulacak kod yapısı ve tasarım, mobil uygulama geliştirme temel prensiplerinin uygulanması ve pekiştirilmesi açısından önemli bir rol oynar.

1. **Var ise XML kodlarının metin olarak yapıştırılması ve açıklanması**

**activity\_main**

**<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <ImageView  
 android:id="@+id/imgLogo"  
 android:layout\_width="100sp"  
 android:layout\_height="100sp"  
 android:contentDescription="Gazi Üniversitesi Logosu"  
 android:src="@drawable/gazi\_logo"  
 tools:layout\_editor\_absoluteX="16dp"  
 tools:layout\_editor\_absoluteY="16dp"  
 tools:ignore="MissingConstraints" />  
  
 <EditText  
 android:id="@+id/etUsername"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintWidth\_percent="0.7"  
 app:layout\_constraintVertical\_bias="0.25"  
 android:hint="Kullanıcı Adı"  
 android:fontFamily="@font/tektur\_font"  
 android:background="@android:drawable/edit\_text" />  
  
 <EditText  
 android:id="@+id/etLocation"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintWidth\_percent="0.7"  
 app:layout\_constraintVertical\_bias="0.35"  
 android:hint="Konum"  
 android:fontFamily="@font/tektur\_font"  
 android:background="@android:drawable/edit\_text" />  
  
 <EditText  
 android:id="@+id/etAge"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintWidth\_percent="0.7"  
 app:layout\_constraintVertical\_bias="0.45"  
 android:hint="Yaş"  
 android:inputType="number"  
 android:fontFamily="@font/tektur\_font"  
 android:background="@android:drawable/edit\_text" />  
  
  
 <Button  
 android:id="@+id/btnSave"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintWidth\_percent="0.7"  
 app:layout\_constraintVertical\_bias="0.6"  
 android:backgroundTint="@android:color/holo\_blue\_dark"  
 android:textColor="@android:color/holo\_blue\_bright"  
 android:text="Kaydet ve Devam Et" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>**

Bu XML dosyası, uygulamanın ana ekranına (Main Activity) ait kullanıcı arayüzünü tanımlamaktadır. Tasarım, ConstraintLayout kullanılarak oluşturulmuş olup, aşağıdaki temel bileşenlerden oluşmaktadır:

1. **Gazi Üniversitesi Logosu (ImageView):**  
   Ekranın sol üst köşesine yerleştirilmiş olan ImageView, 100sp genişliğinde ve yüksekliğinde ayarlanmıştır. Bu alanda Gazi Üniversitesi logosu gösterilmekte ve "Gazi Üniversitesi Logosu" içerik açıklaması ile erişilebilirlik sağlanmaktadır.
2. **Kullanıcı Bilgileri için Giriş Alanları (EditText):**  
   Üç adet EditText, kullanıcıdan sırasıyla “Kullanıcı Adı”, “Konum” ve “Yaş” bilgilerini almak üzere yerleştirilmiştir.
   * Tüm EditText'ler ekranın genişliğinin %70'ini kaplayacak şekilde ayarlanmış ve yatayda ortalanmıştır.
   * Dikey konumlandırma için her bir EditText'e farklı vertical bias değerleri verilmiştir (0.25, 0.35 ve 0.45), bu sayede alanlar birbirlerinden belli aralıklarla yerleştirilmiştir.
   * Kullanıcı dostu bir görünüm için her EditText'e özel font (tektur\_font) ve varsayılan arka plan (edit\_text) uygulanmıştır.
   * “Yaş” alanında, yalnızca sayı girişi yapılabilmesi için inputType özelliği "number" olarak ayarlanmıştır.
3. **Kaydet ve Devam Et Butonu (Button):**  
   Ekranda, kullanıcı tarafından girilen bilgilerin kaydedilmesi ve sonraki ekrana geçişin sağlanması amacıyla bir Button yer almaktadır.
   * Buton da ekranın genişliğinin %70'ini kaplayacak şekilde ayarlanmış ve dikeyde %60 oranında konumlandırılmıştır.
   * Görsel olarak, butonun arka planı koyu mavi tonlarda (holo\_blue\_dark) belirlenirken, metin rengi daha açık bir mavi (holo\_blue\_bright) kullanılarak kontrast oluşturulmuştur.

Genel olarak, bu tasarım kullanıcı deneyimini göz önünde bulundurarak hem işlevsel hem de estetik açıdan uyumlu bir yapı sunmaktadır. Logonun konumlandırılması, kullanıcı bilgilerini almak için yer alan alanlar ve görsel olarak belirgin buton, uygulamanın amacını net bir şekilde ortaya koymaktadır.

**activity\_information**

**<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:orientation="vertical"  
 android:padding="20dp">  
  
 <TextView  
 android:id="@+id/tvUsername"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintWidth\_percent="0.8"  
 app:layout\_constraintVertical\_bias="0.05"  
 android:text="Kullanıcı Adı:"  
 android:textSize="18sp"  
 android:padding="8dp"/>  
  
 <TextView  
 android:id="@+id/tvLocation"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintWidth\_percent="0.8"  
 app:layout\_constraintVertical\_bias="0.25"  
 android:text="Konum:"  
 android:textSize="18sp"  
 android:padding="8dp"/>  
  
 <TextView  
 android:id="@+id/tvAge"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintWidth\_percent="0.8"  
 app:layout\_constraintVertical\_bias="0.45"  
 android:text="Yaş:"  
 android:textSize="18sp"  
 android:padding="8dp"/>  
  
 <EditText  
 android:id="@+id/etDeleteField"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintWidth\_percent="0.8"  
 app:layout\_constraintVertical\_bias="0.65"  
 android:minHeight="48dp"  
 android:padding="12dp"  
 android:hint="Silmek istediğiniz veriyi girin"/>  
  
 <Button  
 android:id="@+id/btnDelete"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintWidth\_percent="0.7"  
 app:layout\_constraintVertical\_bias="0.75"  
 android:text="Veriyi Sil"/>  
  
 <Button  
 android:id="@+id/btnBack"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintWidth\_percent="0.7"  
 app:layout\_constraintVertical\_bias="0.85"  
 android:text="Geri Dön"/>  
  
</androidx.constraintlayout.widget.ConstraintLayout>**

Bu XML dosyası, uygulamanın ikinci ekranına (Bilgi Ekranı) ait kullanıcı arayüzünü tanımlamaktadır. Tasarımda ConstraintLayout kullanılmış olup, ekranın düzeni aşağıdaki bileşenler ve özellikler doğrultusunda oluşturulmuştur:

* **Genel Düzen:**  
  Ekranın tüm kenarlarında 20dp'lik bir padding uygulanmıştır. Layout, bileşenlerin genişliğini ekranın belirli bir yüzdesi olarak ayarlamak için ConstraintLayout’un özelliklerini kullanır.
* **Kullanıcı Bilgilerini Gösteren TextView'ler:**
  + tvUsername: Ekranın üst kısmına yakın (vertical bias 0.05) konumlandırılmış olup, "Kullanıcı Adı:" ifadesini gösterir. Genişliği ekranın %80'ini kaplayacak şekilde ayarlanmıştır ve metin boyutu 18sp ile padding 8dp uygulanmıştır.
  + tvLocation: Bu TextView, vertical bias 0.25 ile ekranın orta üst kısmında yer alır ve "Konum:" metnini gösterir. Genişlik ve stil özellikleri tvUsername ile benzerdir.
  + tvAge: Ekranın orta bölgesinde (vertical bias 0.45) bulunan bu TextView, "Yaş:" ifadesini gösterir ve diğer TextView’lerle uyumlu stil özelliklerine sahiptir.
* **Silme İşlemi İçin EditText:**  
  Ekranın altına doğru, vertical bias 0.65 konumunda yer alan etDeleteField, kullanıcının silmek istediği veriyi (örneğin “username”, “location” veya “age”) girebilmesi için tasarlanmıştır. Minimum yükseklik 48dp, padding 12dp ve açıklayıcı bir hint ("Silmek istediğiniz veriyi girin") bulunmaktadır.
* **Veri Silme Butonu:**  
  btnDelete, kullanıcı tarafından girilen anahtar değerine karşılık gelen veriyi silmek için kullanılacaktır. Bu buton, ekranın %70 genişliğinde olup, vertical bias 0.75 ile yerleştirilmiştir ve "Veriyi Sil" metniyle görsel olarak belirtilmiştir.
* **Geri Dön Butonu:**  
  btnBack, kullanıcıya ilk ekrana geri dönüş imkanı sunar. Buton, yine ekranın %70 genişliğinde ve vertical bias 0.85 konumunda yer almakta, "Geri Dön" ifadesi ile kullanıcıya yönlendirme yapmaktadır.

Genel olarak, bu tasarım kullanıcıya kayıtlı bilgileri net bir şekilde görüntüleme, silme işlemini gerçekleştirme ve uygulama içinde kolayca gezinme imkânı sağlayan sade ve işlevsel bir arayüz sunar.

1. **XML kodları sonucunda ortaya çıkan ekran tasarımının ekran görüntüsü**

**küçük alet, İletişim Cihazı, mobil cihaz, ekran görüntüsü içeren bir resim

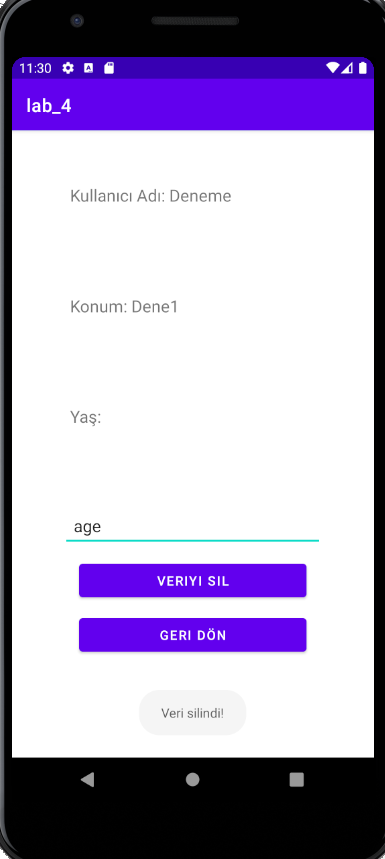
Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**metin, ekran görüntüsü, multimedya, İletişim Cihazı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

****

1. **Var ise Java kodlarının (yorum satırına ek olarak) metin halinde yapıştırılması ve bloklar halinde açıklanması**

**Main\_Activity**

**package com.mobilprogramlama.lab\_4;  
  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class Main\_Activity extends AppCompatActivity {  
 // UI bileşenleri: kullanıcı adını, konumu ve yaşı girmek için EditText alanları  
 private EditText etUsername, etLocation, etAge;  
 // Verileri kaydetmek için buton  
 private Button btnSave;  
 // Uygulama genelinde verileri saklamak için SharedPreferences nesnesi  
 private SharedPreferences sharedPreferences;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 // activity\_main layout dosyasını kullanıcı arayüzü olarak ayarla  
 setContentView(R.layout.*activity\_main*);  
  
 // UI Elemanlarının tanımlanması ve layout dosyasındaki bileşenlerle ilişkilendirilmesi  
 ImageView imgLogo = findViewById(R.id.*imgLogo*); // Uygulama logosunu göstermek için ImageView  
 etUsername = findViewById(R.id.*etUsername*); // Kullanıcı adı girişi  
 etLocation = findViewById(R.id.*etLocation*); // Konum girişi  
 etAge = findViewById(R.id.*etAge*); // Yaş girişi  
 btnSave = findViewById(R.id.*btnSave*); // Verileri kaydetmek için buton  
  
 // "UserData" adında bir SharedPreferences dosyası oluşturulur veya mevcutsa açılır  
 // MODE\_PRIVATE, dosyanın sadece bu uygulama tarafından okunabileceğini belirtir  
 sharedPreferences = getSharedPreferences("UserData", Context.*MODE\_PRIVATE*);  
  
 // Kaydet butonuna tıklama olayını dinleyen (listener) ekleniyor  
 btnSave.setOnClickListener(v -> {  
 // Kullanıcının girdiği metinler alınır  
 String username = etUsername.getText().toString();  
 String location = etLocation.getText().toString();  
 String age = etAge.getText().toString();  
  
 // SharedPreferences üzerinden düzenleme (edit) işlemi başlatılır  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
 // Kullanıcının girdiği veriler, ilgili anahtarlarla kaydedilir  
 editor.putString("username", username);  
 editor.putString("location", location);  
 editor.putString("age", age);  
 // Değişikliklerin kaydedilmesi asenkron olarak yapılır  
 editor.apply();  
  
 // İkinci aktiviteye geçiş yapmak için Intent oluşturulur  
 Intent intent = new Intent(Main\_Activity.this, Second\_Activity.class);  
 // Yeni aktivite başlatılır  
 startActivity(intent);  
 });  
 }  
}**

**Second\_Activity**

**package com.mobilprogramlama.lab\_4;  
  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class Second\_Activity extends AppCompatActivity {  
 // UI bileşenleri: Kayıtlı verileri göstermek için TextView'ler  
 private TextView tvUsername, tvLocation, tvAge;  
 // Kullanıcının silmek istediği veri anahtarını girmek için EditText  
 private EditText etDeleteField;  
 // Veri silme ve geri dönüş işlemleri için butonlar  
 private Button btnDelete, btnBack;  
 // Uygulama genelinde verileri saklamak için SharedPreferences nesnesi  
 private SharedPreferences sharedPreferences;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 // activity\_information layout dosyasını kullanıcı arayüzü olarak ayarla  
 setContentView(R.layout.*activity\_information*);  
  
 // Layout dosyasındaki UI elemanlarının tanımlanması ve ilişkilendirilmesi  
 tvUsername = findViewById(R.id.*tvUsername*); // Kullanıcı adını gösterecek TextView  
 tvLocation = findViewById(R.id.*tvLocation*); // Konumu gösterecek TextView  
 tvAge = findViewById(R.id.*tvAge*); // Yaşı gösterecek TextView  
 etDeleteField = findViewById(R.id.*etDeleteField*); // Silinecek veri anahtarını girmek için EditText  
 btnDelete = findViewById(R.id.*btnDelete*); // Veriyi silmek için buton  
 btnBack = findViewById(R.id.*btnBack*); // İlk ekrana geri dönüş için buton  
  
 // "UserData" adlı SharedPreferences dosyasını oluştur veya aç  
 // MODE\_PRIVATE, dosyanın sadece bu uygulama tarafından okunabileceğini belirtir  
 sharedPreferences = getSharedPreferences("UserData", Context.*MODE\_PRIVATE*);  
  
 // SharedPreferences üzerinden kayıtlı verileri alıp ilgili TextView'lere yazdırma  
 tvUsername.setText("Username: " + sharedPreferences.getString("username", ""));  
 tvLocation.setText("Location: " + sharedPreferences.getString("location", ""));  
 tvAge.setText("Age: " + sharedPreferences.getString("age", ""));  
  
 // Silme işlemi: btnDelete'e tıklanıldığında çalışacak kod bloğu  
 btnDelete.setOnClickListener(v -> {  
 // Kullanıcının EditText'e girdiği silme anahtarını al  
 String deleteKey = etDeleteField.getText().toString();  
 // SharedPreferences üzerinde düzenleme yapmak için editor nesnesi oluştur  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
  
 // Girilen anahtarın "username", "location" veya "age" olup olmadığını kontrol et  
 if (deleteKey.equalsIgnoreCase("username") || deleteKey.equalsIgnoreCase("location") || deleteKey.equalsIgnoreCase("age")) {  
 // Geçerli anahtar ise, ilgili veriyi sil  
 editor.remove(deleteKey);  
 // Değişikliklerin kaydedilmesi (asenkron olarak)  
 editor.apply();  
 // Kullanıcıya verinin silindiğini bildiren Toast mesajı göster  
 Toast.*makeText*(this, "Veri silindi!", Toast.*LENGTH\_SHORT*).show();  
 // Arayüzdeki verileri güncelle  
 refreshData();  
 } else {  
 // Girilen anahtar geçersizse, hata mesajı göster  
 Toast.*makeText*(this, "Geçersiz giriş!", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 // Geri butonuna tıklanıldığında, Main\_Activity'ye dönüş işlemini gerçekleştir  
 btnBack.setOnClickListener(v -> {  
 // Main\_Activity'ye geçiş yapmak için Intent oluştur  
 Intent intent = new Intent(Second\_Activity.this, Main\_Activity.class);  
 startActivity(intent);  
 // Second\_Activity'yi kapat (finish) ederek geri dönüş sağla  
 finish();  
 });  
 }  
  
 // refreshData metodu: SharedPreferences'teki güncel verileri çekerek UI'ı yeniler  
 private void refreshData() {  
 tvUsername.setText("Kullanıcı Adı: " + sharedPreferences.getString("username", ""));  
 tvLocation.setText("Konum: " + sharedPreferences.getString("location", ""));  
 tvAge.setText("Yaş: " + sharedPreferences.getString("age", ""));  
 }  
}**

1. **Kullanılan ek teknolojiler ve yapıların belirtilmesi ve açıklanması**

Bu proje kapsamında, temel Android geliştirme teknolojileri dışında ek yapı ve teknolojiler kullanılarak uygulamanın işlevselliği ve kullanıcı deneyimi artırılmıştır. Aşağıda kullanılan ek teknolojiler ve yapıların açıklamaları yer almaktadır:

* **Android Studio ve Gradle:**  
  Proje, Android Studio IDE kullanılarak geliştirilmiştir. Android Studio, zengin özellikleri ve hata ayıklama araçlarıyla hızlı ve verimli mobil uygulama geliştirme imkânı sunar. Gradle ise proje yapılandırması ve bağımlılık yönetimi için kullanılarak, derleme sürecini otomatikleştirir.
* **Java Programlama Dili:**  
  Uygulamanın iş mantığı, Android SDK ile uyumlu olan Java programlama dili kullanılarak yazılmıştır. Java, nesne yönelimli yapısı ve geniş kütüphane desteği sayesinde, kodun okunabilirliği ve sürdürülebilirliğini sağlamada önemli rol oynar.
* **XML Tabanlı Layout Tasarımı:**  
  Uygulamanın kullanıcı arayüzü, XML dosyaları (activity\_main.xml ve activity\_information.xml) aracılığıyla tanımlanmıştır. Bu yaklaşım, UI bileşenlerinin konumlandırılmasını ve stil ayarlarının merkezi olarak yönetilmesini kolaylaştırır.
  + **ConstraintLayout:**  
    Her iki layout dosyasında da esnek ve duyarlı tasarım imkânı sağlayan ConstraintLayout kullanılarak, UI elemanlarının farklı ekran boyutlarına uyum sağlaması hedeflenmiştir.
* **SharedPreferences:**  
  Kullanıcı verilerinin kalıcı olarak saklanması için Android’in SharedPreferences yapısı kullanılmıştır. Bu yapı, uygulama kapatılıp açılsa bile kullanıcı bilgilerinin korunmasını sağlayarak, veri yönetiminde basit ve etkili bir çözüm sunar.
* **Intent Mekanizması:**  
  Ekranlar arası geçiş işlemleri, Android’in Intent yapısı kullanılarak gerçekleştirilmiştir. Bu sayede, kullanıcı verilerinin kaydedilmesinin ardından farklı aktivitelere sorunsuz bir geçiş sağlanmış, uygulama içi navigasyon kullanıcı dostu hale getirilmiştir.
* **Özel Font ve Stil Ayarları:**  
  Uygulama arayüzünde okunabilirliği artırmak ve estetik bir görünüm sunmak amacıyla özel font (örneğin, “tektur\_font”) kullanılmıştır. Renk, padding ve diğer stil ögeleri, kullanıcı deneyimini geliştirmek için özenle belirlenmiştir.

Bu ek teknolojiler ve yapılar, projenin hem teknik açıdan sağlam hem de kullanıcı dostu bir mobil uygulama olarak ortaya çıkmasını sağlamaktadır. Hem veri yönetiminde hem de arayüz tasarımında kullanılan bu teknolojiler, modern Android uygulama geliştirme standartları ile uyumlu bir yapı sunar.