**DATABASE CLASSI**

**package** com.example.serdar.ikazet;

**import** android.content.ContentValues;
**import** android.content.Context;
**import** android.database.Cursor;
**import** android.database.sqlite.SQLiteDatabase;
**import** android.database.sqlite.SQLiteOpenHelper;

**import** java.util.HashMap;

**public class** Database **extends** SQLiteOpenHelper {

 *// All Static variables
 // Database Version* **private static final int *DATABASE\_VERSION*** = 1;

 *// Database Name* **private static final** String ***DATABASE\_NAME*** = **"id224792\_wesdesa"**;*//database ad�* **private static final** String ***TABLE\_NAME*** = **"diyosantable"**;
 **private static** String *KULLANICI\_ID* = **"id"**;
 **private static** String *KULLANICI\_EMAIL* = **"Email"**;
 **private static** String *KULLANICI\_SIFRE* = **"Sifre"**;
 **private static** String *KAYIT\_TARIHI* = **"Tarih"**;

 **public** Database(Context context) {
 **super**(context, ***DATABASE\_NAME***, **null**, ***DATABASE\_VERSION***);
 }

 @Override
 **public void** onCreate(SQLiteDatabase db) { *// Databesi olu�turuyoruz.Bu methodu biz �a��rm�yoruz. Databese de obje olu�turdu�umuzda otamatik �a��r�l�yor.* String CREATE\_TABLE = **"CREATE TABLE "** + ***TABLE\_NAME*** + **"("** + *KULLANICI\_ID* + **" INTEGER PRIMARY KEY,"** + *KULLANICI\_EMAIL* + **" TEXT,"** + *KULLANICI\_SIFRE* + **" TEXT,"** + *KAYIT\_TARIHI* + **" TEXT"** + **")"**;
 db.execSQL(CREATE\_TABLE);
 }

 **public void** kullaniciEkle(String Email,String Sifre,String Tarih) {
 *//kullan�c�Ekle methodu ise ad� �st�nde Databese veri eklemek i�in* SQLiteDatabase db = **this**.getWritableDatabase();
 ContentValues values = **new** ContentValues();
 values.put(*KULLANICI\_EMAIL*, Email);
 values.put(*KULLANICI\_SIFRE*, Sifre);
 values.put(*KAYIT\_TARIHI*, Tarih);

 db.insert(***TABLE\_NAME***, **null**, values);
 db.close(); *//Database Ba�lant�s�n� kapatt�k\*/* }

 **public** HashMap<String, String> kullaniciDetay(){
HashMap<String,String> kisi = **new** HashMap<String,String>();
 String selectQuery = **"SELECT \* FROM "** + ***TABLE\_NAME***;

 SQLiteDatabase db = **this**.getReadableDatabase();
 Cursor cursor = db.rawQuery(selectQuery, **null**);

 cursor.moveToFirst();
 **if**(cursor.getCount() > 0){
 kisi.put(*KULLANICI\_EMAIL*, cursor.getString(1));
 kisi.put(*KULLANICI\_SIFRE*, cursor.getString(2));
 kisi.put(*KAYIT\_TARIHI*, cursor.getString(3));
 }
 cursor.close();
 db.close();
 *// return kitap* **return** kisi;
 }

 **public int** getRowCount() { *//tabloda ka� sat�r kay�tl� oldu�unu geri d�ner* String countQuery = **"SELECT \* FROM "** + ***TABLE\_NAME***;
 SQLiteDatabase db = **this**.getReadableDatabase();
 Cursor cursor = db.rawQuery(countQuery, **null**);
 **int** rowCount = cursor.getCount();
 db.close();
 cursor.close();
 *// return row count* **return** rowCount;
 }

 **public void** resetTables(){
 *// T�m verileri siler. tabloyu resetler.* SQLiteDatabase db = **this**.getWritableDatabase();
 db.delete(***TABLE\_NAME***, **null**, **null**);
 db.close();
 }

 @Override
 **public void** onUpgrade(SQLiteDatabase arg0, **int** arg1, **int** arg2) {
 *//* ***TODO Auto-generated method stub*** }

}

FONKSIYONLAR CLASSI

**package** com.example.serdar.ikazet;

**import** android.content.Context;

**import** java.security.MessageDigest;
**import** java.security.NoSuchAlgorithmException;
**import** java.util.regex.Matcher;
**import** java.util.regex.Pattern;

**public class** Fonksiyonlar {

 **public static boolean** isEmailValid(String Email) { *//mail format� kontrol eder* **boolean** isValid = **false**;

 String expression = **"^[\\w\\.-]+@([\\w\\-]+\\.)+[A-Z]{2,4}$"**;
 CharSequence inputStr = Email;

 Pattern pattern = Pattern.*compile*(expression, Pattern.***CASE\_INSENSITIVE***);
 Matcher matcher = pattern.matcher(inputStr);
 **if** (matcher.matches()) {
 isValid = **true**;
 }
 **return** isValid;
 }

 **public static** String sha1(String data) *//Sha1 �ifreleme yapar* {
 **try** {
 **byte**[] b = data.getBytes();
 MessageDigest md = MessageDigest.*getInstance*(**"SHA-1"**);
 md.reset();
 md.update(b);
 **byte** messageDigest[] = md.digest();
 StringBuilder result = **new** StringBuilder();
 **for** (**int** i = 0; i < messageDigest.**length**; i++)
 {
 result.append(Integer.*toString*((messageDigest[i] & 0xff) + 0x100, 16).substring(1));
 }

 **return** result.toString();

 } **catch** (NoSuchAlgorithmException e)
 {

 *// Log.e("ARTags", "SHA1 is not a supported algorithm");* }
 **return null**;
 }

 **public static boolean** giriskontrol(Context context){
 Database db = **new** Database(context);
 **int** count = db.getRowCount();*// databasedeki table logindeki row say�s�* **if**(count > 0){*//0 dan fazla ise giri� yapm�s �nceden demek
 //kullan�c� giri� yapmıs* **return true**;
 }
 **return false**;
 }

}

POST CLASSI

**package** com.example.serdar.ikazet;

**import** android.util.Log;

**import** org.apache.http.HttpEntity;
**import** org.apache.http.HttpResponse;
**import** org.apache.http.NameValuePair;
**import** org.apache.http.client.ClientProtocolException;
**import** org.apache.http.client.entity.UrlEncodedFormEntity;
**import** org.apache.http.client.methods.HttpGet;
**import** org.apache.http.client.methods.HttpPost;
**import** org.apache.http.client.utils.URLEncodedUtils;
**import** org.apache.http.impl.client.DefaultHttpClient;
**import** org.apache.http.params.BasicHttpParams;
**import** org.apache.http.params.HttpConnectionParams;
**import** org.apache.http.params.HttpParams;

**import** java.io.BufferedReader;
**import** java.io.IOException;
**import** java.io.InputStream;
**import** java.io.InputStreamReader;
**import** java.io.UnsupportedEncodingException;
**import** java.util.List;

**public class** PostClass {
 **static** InputStream *veri*;
 **static** String *veri\_string*;

 **public** PostClass() {
 *//* ***TODO Auto-generated constructor stub*** }

 **public** String httpPost(String url, String method,List<NameValuePair> params,**int** time) {

 **try** {

 **if** (method == **"POST"**) {

 HttpParams httpParameters = **new** BasicHttpParams();
 **int** timeout1 = time;
 **int** timeout2 = time;
 HttpConnectionParams.*setConnectionTimeout*(httpParameters, timeout1);
 HttpConnectionParams.*setSoTimeout*(httpParameters, timeout2);
 DefaultHttpClient httpClient = **new** DefaultHttpClient(httpParameters);
 HttpPost httpPost = **new** HttpPost(url);
 httpPost.setEntity(**new** UrlEncodedFormEntity(params,**"utf-8"**));
 HttpResponse httpResponse = httpClient.execute(httpPost);
 HttpEntity httpEntity = httpResponse.getEntity();
 *veri* = httpEntity.getContent();

 } **else if** (method == **"GET"**) {

 DefaultHttpClient httpClient = **new** DefaultHttpClient();
 String paramString = URLEncodedUtils.*format*(params, **"utf-8"**);
 url += **"?"** + paramString;
 HttpGet httpGet = **new** HttpGet(url);

 HttpResponse httpResponse = httpClient.execute(httpGet);
 HttpEntity httpEntity = httpResponse.getEntity();
 *veri* = httpEntity.getContent();
 }

 } **catch** (UnsupportedEncodingException e) {
 e.printStackTrace();
 } **catch** (ClientProtocolException e) {
 e.printStackTrace();
 } **catch** (IOException e) {
 e.printStackTrace();
 }
 **try** {
 BufferedReader reader = **new** BufferedReader(**new** InputStreamReader(
 *veri*, **"iso-8859-1"**), 8);
 StringBuilder sb = **new** StringBuilder();
 String line = **null**;
 **while** ((line = reader.readLine()) != **null**) {
 sb.append(line + **"\n"**);
 }
 *veri*.close();
 *veri\_string* = sb.toString();
 } **catch** (Exception e) {
 Log.*e*(**"Buffer Error"**, **"Hata "** + e.toString());
 }

 **return** *veri\_string*; *// Ald���m�z cevab�n string halini geri d�n�yoruz* }

}