public class DBConnection extends SQLiteOpenHelper {

private final static int DATABASE\_VERSION = 1;

private final static String DATABSE\_NAME = "smartdeviceappdevelopment.db";

public DBConnection(@Nullable Context context) {

super(context, DATABSE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

String sql = "CREATE TABLE " + RegistrationEntity.TABLE\_NAME + "(" +

RegistrationEntity.\_ID + " INTEGER PRIMARY KEY," +

RegistrationEntity.FULL\_NAME + " TEXT," +

RegistrationEntity.USER\_NAME + " TEXT," +

RegistrationEntity.PASSWORD + " TEXT," +

RegistrationEntity.EMAIL\_ADDRESS + " TEXT," +

RegistrationEntity.CONTACT\_NUMBER + " TEXT)";

db.execSQL(sql);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL(RegistrationEntity.TABLE\_UPGRADE);

onCreate(db);

}

// store data in sql database

public void addRegistrationData(String fullName, String userName, String password, String email, String contact) {

SQLiteDatabase database = getWritableDatabase();

ContentValues registrationDataSave = new ContentValues();

registrationDataSave.put(RegistrationEntity.FULL\_NAME, fullName);

registrationDataSave.put(RegistrationEntity.USER\_NAME, userName);

registrationDataSave.put(RegistrationEntity.PASSWORD, password);

registrationDataSave.put(RegistrationEntity.EMAIL\_ADDRESS, email);

registrationDataSave.put(RegistrationEntity.CONTACT\_NUMBER, contact);

long resultOfRegistration = database.insert(RegistrationEntity.TABLE\_NAME, null, registrationDataSave);

if (resultOfRegistration > 0) {

System.out.println("Data save successfully");

}

}

// check or search data in sql database

public boolean checkLogin(String username, String password) {

try {

SQLiteDatabase db = getReadableDatabase();

String[] projection = {

RegistrationEntity.\_ID

};

String selection = RegistrationEntity.USER\_NAME + "=? AND " +

RegistrationEntity.PASSWORD + "=?";

String[] selectionArgs = {username, password};

Cursor cursor = db.query( //data structure:data access korte help kore

RegistrationEntity.TABLE\_NAME,

projection, //table colum:id,name

selection,

//search korar somoy single colums select kore search kora holo selection like id=1?or name=sam Ekhane akhon id,name koyekta selection

selectionArgs,

//selection er value like id=1,ekhane 1holo selectn args

null,

null,

null);

if (cursor != null && cursor.getCount() > 0) {

while (cursor.moveToNext()) {

int id = cursor.getInt(

cursor.getColumnIndexOrThrow(

RegistrationEntity.\_ID));

if (id > 0)

return Boolean.TRUE;

}

}

} catch (Exception ex) {

Log.e("DBERROR", ex.getMessage());

}

return Boolean.FALSE;

}

public boolean updateData(String name, int id) {

SQLiteDatabase db = getWritableDatabase();

ContentValues contentValues = new ContentValues();

contentValues.put(RegistrationEntity.FULL\_NAME, name);

String selection = RegistrationEntity.\_ID + "=?";

String[] selectionArgs = {String.valueOf(id)};

int updateResult = db.update(RegistrationEntity.TABLE\_NAME, contentValues, selection, selectionArgs);

if (updateResult > 0)

return Boolean.TRUE;

return Boolean.FALSE;

}

public boolean deleteData(int id) {

SQLiteDatabase db = getWritableDatabase();

String selection = RegistrationEntity.\_ID + "=?";

String[] selectionArgs = {String.valueOf(id)};

int deleteResult = db.delete(RegistrationEntity.TABLE\_NAME, selection, selectionArgs);

if (deleteResult > 0)

return Boolean.TRUE;

return Boolean.FALSE;

}

}