Name : Rishi Ram

Roll no : 20230201032

Page No. 1

Date 26 09 2024

Assignment 4

Q.1 Explain cursor in android with suitable example

Ans

In Android, a cursor is an interface that provides random read-write access to the result set returned by a database query.

Cursors are typically used when interacting with databases, such as SQLite databases or content providos to efficiently retrive & manipulate data.

Key features of a cursor

1. Positioning - Cursor allows you to traverse the row of a database query result set one row at a time.
2. ~~Column~~ access - You can retrieve data from individual data from individual column using methods like get String () getInt (), getDouble(),
3. Size Information - A cursor provides information about the size of the result set & whether it contains any data

eg.

Demonstrate how a cursor is used to fetch data from SQLite database & display it in a ListView

1. Setting up the SQLite database

```
public class MyDatabaseHelper extends SQLiteOpenHelper{
    private static final String
        DATABASE_NAME ="mydatabase.db";
    private static final int DATABASE_VERSION=1;
    public MyDatabaseHelper(context context){
        super(context,DATABASE_NAME,null,
                DATABASE_VERSION);
    }
@ Override
public void onCreate (SQLiteDatabase db) {
    db.execSQL ("CREATE TABLE items (_id
 INTEGER PRIMARY KEY AUTOINCREMENT,
 name. Text, price REAL)";");
 db.execSQL ("INSERT INTO items (name, price)
 VALUES ("Apple, 1.99);");
  db.execSQL ("INSERT INTO items (name, price)
  VALUES ("Banana"; 0.99);");
 }
 @ Override
 public void onUpgrade (SQLiteDatabase db,
     int oldversion, int newVersion) {
 db.execSQL ("DROP TABLE IF EXISTS items");
 onCreate (db);
 }
}
```

2. Querying data with a cursor

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
   private MyDatabaseHelper dbHelper;
   private ListView listView;
   @ Override
   protected void onCreate (Bundle savedInstanceState)
   {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_main);
     listView = findViewById(R.id.listView);
     dbHelper = new MyDatabaseHelper (this);
```

```
SQLiteDatabase db = dbHelper.getReadableDatabase();
Cursor cursor = db.rawQuery ("SELECT _id,
    name, price FROM items", null);
SimpleCursorAdapter adapter = new SimpleCursor
Adapter. this;
android.R.layout.simple_list_item_2;
cursor,
new String [] {"name", "price"}
new Int [] { android.R.id.text1,
                android.R.id.text2}, 0);
list View.setAdapter (adapter); }}
```

3. Layout File (activity_main.xml)

```
<? xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android
.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">

<ListView
android:id="@+id/listView"
android:layout_width="match_parent"
android:layout_height="match_parent"/>

</LinearLayout>
```

Q.2 Write short note on SQLite databas

Ans

SQLite is a lightweight, relational database engine that is embedded within Android devices. It is used to store structured data in databases & provides a powerful, yet compact, SQL based interface for managing data. SQLite databases are self contained serverless & do not require separate installations, making them ideal for mobile environments where storage & memory resources are limited.

Key features & suite in android.

- lightweight & embedded
- SQL syntax
- Data storage
- Persistent storage
- offline access

Q.3 Write short note on Content value with example

Ans

Content values is a key value pair class used to store & pass data in android, specifically when interacting with databases like SQLite or content providers. It allows developers to insert or update data in a database table without needing to write SQL statements directly.

Each entry in content values represents a column name & corresponding value for that column. It is commonly used when performing INSERT, UPDATE, DELETE operation in SQLite database.

Inserting data into SQLite database any content values.

```
My Database Helper dbHelper = new myDatabaseHelper
                                   (context);

SQLiteDatabase db = dbHelper.getVariableDatabase();
ContentValues values = new ContentValues();

values.put("name", "apple");
values.put("price", 1.99);

Long newRowId = db.insert("items", null, values);
db.close();
```

Assignment 4 - Q1

Database Connectivity

DBHelper.java - Java File

```
package com. example . sqlitecrud;

import ...

public class DBHelper extends SQLite OpenHelper{
    public static Final string Database="student.db";
    public static final string table_name="student";
    public static final string id="ID";
    public static final string name ="Nane";
    public static final string surname ="SURNane";
    public static final string marks="MARKS";

    public DBHelper (Context context){
        super(context. Database, null ());
    }
    @Override
    public void onCreate (SQLite Database db){
        db.exec SQL ("Create table" + table_name+
            (ID Integer primary key. name text,
            surname text, marks text)");
    }

    @Override
    public void onUpgrade (SQLiteDatabase db, int
        oldversion, int newVersion){
```

```
    db.execSQL("DROP TABLE IF EXISTS "+
table_name) onCreate(db);
    }

  public boolean insertData(String name, String
      dsurname, String dmarks){
      SQLiteDatabase db= this.getWritableDatabase();
      ContentValues values= new ContentValues();
      values.put(name, dname);
      values, put(surname, dsurname);
      values.put(marks, dmarks);
      long result= db.insert(table_name, null, value)
      return result!=-1;
  }

  public Cursor getAllData(){
      SQLiteDatabase db= this.getWritableDatabase();
      return db.rawQuery("SELECT * FROM " + 
             table_name, null);
  }

  public boolean updateData(String id, String name,
        String surname, String marks){
     SQLiteDatabase db= this.getWritableDatabase();
     ContentValues values= new ContentValues();
     values.put(id;id);
     dbupdate(table_name, values, "ID=?" new
          String[] {id});
     return true;
  }
```

```
  public Integer deleteData(String id){
     SQLiteDatabase db= this.getWritableDatabase();
     return db.delete(table_name, "ID=?",
          new String[] {id});
  }
}
```

MainActivity.java - Java File

```
package com.example.database_connectivity;

import...

public class MainActivity extends AppCompatActivity{
    DBHelper mydb;
    @Override
    protected void onCreate(Bundle savedInstance){
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
       mydb = new DBHelper(this);

       EditText name=findViewById(R.id.name);
       EditText sname=findViewById(R.id.sname);
       EditText marks=findViewById(R.id.marks);
       EditText id=findViewById(R.id.id);
       EditText add=findViewById(R.id.add);
       EditText update=findViewById(R.id.update);
       EditText view=findViewById(R.id.view);
       EditText delete=findViewById(R.id.delete);
```

```
  add.set OnClickListener(new View.On ClickListner(){
     @Override
     public void onClick(View view){
          boolean isInserted.mydb.insertData(name.
getText().to string(), sname.getText().to String();
marks.get Text().to String());
     }
});


  View.set OnClickListener(new View.OnClickListener(){
     @Override
     public void on Click(View view.){
        cursor res = mydb.getAllData();
        if(res.getCount()==0){
            showMessage("Error,"Nothing found");
        }
        while(re).moveToNext()){
            buffer.append(res.getString(0)+"\n");
            buffer.append(res.getString(1)+"\n");
            buffer.append(res.getString(2)+"\n");
            buffer.append(res.getString(3)+"\n\n");
        }
        show Message("Data", buffer.to String());
     }
  });


delete.setOnClickListener(new View.on ClickListener{
   @Override
   public void onClick(View view){
        mydb.deleteData(id.getText().toString())
   }
});
```

```
   update.set OnClick Listener(new view.onClickListener){
      @Override
      public void onClick(View view){
           boolean isUpdated = mydb.UpdateData(id.
getText().to String(), name.getText().toString(),
Sname.getText().toString(), marks.getText.toString());

           if(isUpdated==true){
              Toast.makeText(this,"Data Updated",
                 Toast.LENGTH_LONG).show();
           }else{
              Toast.makeText(this," Data not updated
                 Toast.LENGTH-LONG).show();
           }
     });
}


 public void show Message(string title, string messg){
    AlertDialog.Builder builder=new AlertDialog(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(messg)
    builder.show();
  }
}
```

activity_main.xml - XML File

```
<RelativeLayout xmlns:android="http://schemas.
android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="16dp">

<EditText
android:id="@+id/id"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="ID" />

<EditText
android:id="@+id/name"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="name" />

<EditText
android:id="@+id/surname"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="surname"/>

<EditText
android:id="@+id/marks"
android:layout_height="match_parent"
android:layout_width="wrap_content"
android:hint="marks" />
```

```
<~~EditText~~ Button
android:id="@+id/add"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Add" />

<Button
android:id="@+id/view"
android:layout_width="match_parent"
android:layout_height="wrap_content" />

<Button
android:id="@+id/update"
android:layout_width="match_parent"
android:layout_height="wrap_content" />

<Button
android:id="@+id/Delete"
android:layout_width="match_parent"
android:layout_height="wrap_content" />

</RelativeLayout>
```

Assignment 4

webView

MainActivity.java - Java File

```
package com.example.webview;

import ...

public class MainActivity extends AppCompatActivity{
    @Override
     protected void onCreate(Bundle savedInstanceState)
          super.onCreate(savedInstanceState);
          setContentView(R.layout.activity_main);

          EditText search= findViewById(R.id.edurl);
          ImageButton url= findViewById(R.id.url);
          WebView wv= FindViewById(R.id.wv);

          url.setOnClickListener(new View.onClick
Listener(){
               @Override
               public void onClick(View v){
                    String s= search.getText().toString();
                    wv.loadurl(s);
               }
          });
     }
}
```

activity_main.xml - XML File

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/
android.com"
xmlns:app="http://schemas.android.com/app/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
/>

<EditText
android:id="@+id/edurl"
android:layout_width="match_parent"
android:layout_height="wrap_content"
/>

<WebView
android:id="@+id/webview"
android:layout_width="match_parent"
android:layout_Height="wrap_content"
/>

</RelativeLayout>
```