SMVS
Page No.
Date : / /

Android Storages APIs

* Explain Shared Preference with Example.

=> Shared Preference in Android are used to store and retrieve- key value pairs of primitive data types across whole application sessions.

They are commonly used to store user preferences, settings and other small amounts of data.

Shared Preference provide a way to store simple data.

They are stored as an XML File in the app's private storage.

Shared Preference is used to add, delete and update preference in Android application.

Page No.
Date : / /

1 Adding a Preference:

Used to add the Preference in Android Application.

Use 'sharedPreferences.Editor' to add a preference.

Example:

```
SharedPreferences sharedprefrences =
    getSharedPreferences("MyPrefs",
        Context.MODE_PRIVATE);

SharedPreferences.Editor
    editor = sharedPreferences.edit();

editor.putBoolean("isDarkMode",
        true);
editor.apply();
```

2 Updating Preference:

Used to update the Preference in Arodroid Application.

Example:

SMVS
Page No.
Date : / /

```
editor.putBoolean("isDarkMode", false);
editor.apply();
```

3 Deleting a Preference:

Used to remove to delete a specific preference and clear to delete all preference.

Example:

```
editor.remove("isDarkMode");
editor.apply();
editor.clear();
editor.apply();
```

=> Example:

```
public class MainActivity extends
    AppCompatActivity
{
    private EditText et;
    private TextView tv;
    private SharedPreferences
            SharedPreferences;

    protected void onCreate()
    {
    et=FindViewById(R.id.ET);
```

```
tv = FindViewById(R.id.TV);

SharedPreference = getSharedPreference
  ("MyPrefs", MODE_PRIVATE);

FindViewById(R.id.saveButton).
setOnClickListener(new View.
  OnClickListener()
{
  public void onClick(View v)
  {
    String name = et.getText().
      toString().trim();
    if(!name.isEmpty())
    {
      sharedPreferences.edit().
        putString("username", name).
          apply();
      tv.setText("Hello, " +name);
    }
  }
});

String savedname = sharedPreference
 .getString("username", "");

tv.setText("Hello, ", savedname);

}
}
```

SMVS
Page No.
Date : / /

* Explain Simple Timer Application.

=>
```
public class MainActivity extends
  AppCompatActivity
{
  private TextView tv;
  private Button btn;
  private CountDownTimer timer;

  protected void onCreate()
  {
    tv = FindViewById(R.id.TV);
    btn = FindViewById(R.id.button);

    btn.setOnClickListener(new
      View.OnClickListener()
    {
      public void onClick(View v)
      {
        startTimer();
      }
    });
  }
  private void startTimer()
  {
    timer = new CountDownTimer
            (3000, 1000)
  {
```

SMVS
Page No.
Date : / /

```
    public void onTick(long millisUntil
        Finished)
    {
      tv.setText("seconds Remaining:"
        + millisUntilFinished / 1000);
    }
    public void onFinish()
    {
      tv.setText("Time is Up!");
      Toast.makeText(MainActivity.
        this, "Timer Finished",
        Toast.LENGTH_SHORT).show();
    }
  }.start();
}
protected void onDestroy()
{
  super.onDestroy();

  if(countDownTimer != null)
  {
    timer.cancel();
  }
}

}
```

SMVS
Page No.
Date : / /

* Explain SQlite Database in Android.

=> SQlite is a lightweight, Relational Database Management System which is widely used in mobile and desktop application.

It provides a simple way to store, manage and retrieve structured data.

=> Steps For Create Database:

1 Initialize SQLiteOpenHelper:

Extend the 'SQLiteOpenHelper' class to manage database creation.

```
public class MyDatabaseHelper
  extends SQLiteOpenHelper
{

    // code

}
```

Page No.
Date : / /

2 Create and Open Database Instance:

Use 'MyDatabaseHelper' class to create or open a connection to the SQLite Database.

```
MyDatabaseHelper dbHelper =
    new MyDatabaseHelper(context);

SQLiteDatabase db = dbHelper.
    getWritableDatabase();
```

3 Updating and Deleting Database Records.

Once your database connection is create, then you can perform ea CRUD operation.

(i) Inserting Records:

```
ContentValues values = new
        ContentValues();

values.put("name", "BrainSpot");

long newId = db.insert("my_table",
    null, values);
```

SMVS
Page No.
Date : / /

(ii) Updating Records:

```
ContentValues updatedvalues =
    new ContentValues();

updatedvalues.put("name", "Brain");

int rowsAffected = db.update("my-table",
    UpdatedValues, "id=?", new string
    [] {"1"});
```

(iii) Deleting Records:

```
int deletedrows = db.delete("my-table",
    "id?", new String[] {"1"});
```

4 Closing and Deleting SQLite Database:

```
db.close();

Context.deleteDatabase
    ("mydatabase.db");
```

SMVS
Page No.
Date : / /

=> Example:

-> MyDatabaseHelper.java:

```
public class MyDatabaseHelper
  extends SQLiteOpenHelper
{
  private static final String
    DATABASE_NAME = "mydb.db";
  private static final int
    DATABASE_VERSION = 1;

  public void onCreate(SQLite
    Database db)
  {
    db.execSQL("CREATE TABLE IF
      NOT EXISTS my_table( id
        INTEGER PRIMARY KEY,
        name TEXT)");
  }

  public void insertData(String
    name)
  {
    SQLiteDatabase db = this.
      getWritableDatabase();
    ContentValues values = new
        ContentValues();
    values.put("name", name);
```

```
        long newRowId = db.insert("
            my_table", null, values);
        db.close();

        return newRowId;
    }
}
```

-> MainActivity.java :

```
public class MainActivity extends
    AppCompatActivity
{
    protected void onCreate()
    {
        MyDatabaseHelper dbHelper = new
            MyDatabaseHelper(this);

        long newRowId = dbHelper.insertData
                ("Brain Spot");

        if(newRowId != -1)
        {
            Toast.makeText(this, "Data
                inserted Done", Toast.
                LENGTH_SHORT).show();
        }
    else
```

SMVS
Page No.
Date : / /

```
        {
            Toast.makeText(this, "Failed
              to Insert", Toast.LENGTH_
              SHORT).show();
        }
    }
}
```

* Explain Simple Counter App:

=>
```
public class MainActivity extends
    AppCompatActivity
{
    private int counter = 0;
    private TextView tv;
    private Button btn;

    protected void onCreate()
    {
        tv = FindViewById(R.id.TV);
        btn = FindViewById(R.id.button);

        btn.SetonClickListener(new
          View.OnClickListener()
        {
            public void onclick(View v)
            {
                Counter ++;
```

SMVS
Page No.
Date : / /

```
        tv.setText(String.valueOf
                (counter));
    }
});

}
```