Assignment - 7

(Q-1) Based on your understanding, identify a recent business trend that has influced the Android Platform. Explain how this trend impacts Android App developers and businesses in the mobile app industry.

→ A recent business trend that has significantly influenced the Android platform is the rise of 5G technology. The widespread 5G networks has had a substantial impact on Android app developers and businesses in the app mobile app industry

→ With 5G, Android app developers are now able to create more advanced and data-intensive applications. This technology offers significantly faster download and upload speeds, lower latency and enhanced network reliability

→ Businesses can develop innovative apps that leverage the capabilities of 5G networks, such as Augmented Reality (AR) and virtual Reality (VR) applications, high-definition video streaming and gaming experiences.

→ The rise of 5G technology on the Android platform empowers developers to create more advanced and impresive apps, while businesses can explore innovative ideas, enhance user engagment and drive revenue growth through data-driven strategies

Q-2) What is the purpose of an inflator of layout in Android platform, and how does it fit into the architecture of Android layouts?

→ In Android development an inflater is a crucial component used to convert an XML layout file into corresponding view objects in main memory.

→ It is commonly used a dynamically create and populated the UI (User Interface) of an Android application.

- Here's how Inflator layout fits into the architecture:

1) XML layout file: Android developers define the UI of their app's activity and fragments using XML layout files. These layout files describe the structure and apperance of the UI elements such as textview, Button and containers like LinearLayout, ConstraintLayout etc. XML file located in and stored in res/layout.

2) Layout Inflation: when an activity or fragment is created or needs to display a specific UI layout, the XML layout file associated with that UI is inflated.

3) Inflating with layoutInflater: To perform layout inflation, developers typically use an instance of the 'LayoutInflater' class.

4) Populating and Managing view: Once the XML layout is inflated you can access and manipulate the individual view objects it contains

5) Displaying the UI: Finally, the inflated view hierarchy can be added to the activity's content view or fragment's layout using methods like 'setContentView()'. This step makes the UI visible to the User.

Q-3) Explain the concept of a CustomDialogBox in Android development. Provide examples to illustrate its use.

A custom dialog box in Android is a user interface element that allows developers to create a customized popup or model dialog to interact with users.

simple example of creating and using a custom dialog in Android

```
fun showCustomDialog() {
    val customDialog = Dialog(this)
customDialog.setContentView(R.layout.custom_dialog)
val messageTextView = customDialog.findViewById<TextView>
                        (R.id.messageTextView)
val okButton = customDialog.findViewById<Button>(R.id.okButton)

messageTextView.text = "This is a custom dialog!"
okButton.setOnClickListner {
                customDialog.dismiss()}
customDialog.show()
```

Q-4) How do activities, services and the Android Manifest file work together to make an Android app? Can you describe their main roles and provide a basic how they cooperate to design a mobile app?

Roles:

1) Activities:

Role: Activities represent the user interface (UI) and user interaction. Each screen or UI component of an app is typically implemented as an activity.

2) Services:

Role: Services are background components that perform long-running operations without a visible UI. They run independently of activities and are used for tasks such as playing music, downloading etc.

3) Android Manifest File.

Role: The android manifest file provides essential information about the app to the android system. It serves as a configuration file that specifies the app's component, permissions and other critical details

~~# Android~~

let's take example of TO-DO list app

~~## Android manifest~~

Example

```
Class MainActivity : AppCompactActivity() {
override fun onCreate(savedInstanceState : Bundle?) {
super.onCreate(saveInstanceState)
setContentView(R.layout.activity_main)
startServiceButton.setOnClickListner {
Val startService serviceIntent = Intent(this,
NotificationService::class.java)
startService(serviceIntent)
}
}
}

Class NotificationService : IntentService("NotificationService") {
override fun onHandleIntent(intent: Intent?) {
if(intent != null) {
createNotification()}}

Private fun createNotification() {
Val channelId = "my_channel"
if(Build.Version.SDK_INT >= Build.version_codes.O) {
Val name = "my_channel"
Val notificationManager = getsystemservice(NotificationManager::
class.java)
INotificationManager.createNotificationChannel
```

```
Val Builder = Notification.Compat.Builder (this, channelID).
    .setSmallIcon(R.Drawable.IC_launcher_foreground).
    .setContentText("This is notification from service").
```

Q-5) How does the Android Manifest file impact the development of an Android application? Provide an example to demonstrate its significant.

The Android Manifest file is a crucial component of an Android application. It serve as a central configuration file that provides essential information about the app to the Android operating system.

1) Defining App Components

2) Permissions

3) App MetaDatu

4) Intent Filters

5) Activity Launchers.

Ex > If you are developing a messaging app.,

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    Package="com.example.messagingapp">

    <application
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme>

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER/>
            </intent-filter>
        </activity>

        <user-Permission android:name="android.Permission.INTERNET"/>
        <user-Permission android:name="android.Permission.SEND_SMS"/>
        <user-Permission android:name="android.permission.READ_CONTACTS"/>

    </applications>
    //manifest>
```

Q-6) What is the role of resources in Android development ? Discuss the various types of resource and their significance in creating well-structured application. Provide example to clarify your points.

→ Resources plays a crucial role in android development as they provide way to seperate the app's code from it's assets and content, enabling developers to create flexible, maintainable and user-friendly applications. Android resources includes various types --

1) Layout Resource.

type : XML files found in 'res/layout' directory
significance : Define the structure and appearance of the UI

Ex?

```
<LinearLayout
  android:layout_width = "match_parent"
  android:layout_height = "match_parent"
  android:orientation = "vertical">

  < TextView
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:text = "Vyom_Prajapati_22172012031" />

</LinearLayout>
```

2) Drawable Resource:

type: Image files (e.g. JPEG, PNG) found in 'res/drawable'.
Significance: Store images and icons that we can use in app's UI.

Ex:
res/drawable/ic_app_logo.png

3) String Resource:

type: XML files in 'res/value' directory
Significance: Store text strings used in the app.

Ex
```
<resources>
    <string name = "app_name" >myApp </string>
</resources>
```

4) Color Resources:

type: XML files in the 'res/value" directory
Significance: Store color value for theming and styling of the app.

Ex:
```
<resources>
    <color name = "color Primary" >#FF0000 </color>
```

Q-7) How does an Android service contribute to the functionality of a mobile application? Describe the process of developing an Android service.

- An Android service is a fundamental component that contributes to the functionality of a mobile application by allowing background tasks and long-running processes to execute independently of the app's UI.

→ Services are essential for tasks that need to continue running even mean the app is not in the foreground, such as playing ~~music~~ music, downloading data etc.

- Android service functionalities.

1) Background Processing
2) Long-Running tasks
3) Inter Component Communication
4) Foreground Services.

Developing Process of Android Services:

1) Create a service class

```
class MyService : Service() {
override fun onStartCommand (Intent : Intent?, flags : Int,
                             startId : Int) : Int {
    // Perform background tasks here
    return START_STICKY
}
override fun onBind( intent: Intent? ): IBinder? {
    return null
}
}
```

2) Define the service in the manifest

```
< service android:name=".myservie" />
```

3) Start and stop service: you can start a service using an "Intent" and stop it when it's no longer needed

```
Val intent= Intent (this, Myservice :: class.java)
Startservice( Intent)

Stopservice( Intent)
```