

Assignment :- 1

Q.1 Based on your understanding, identify & recent business trend that has influenced the Android platform. Explain how this trend impacts Android app developers & businesses in the mobile app industry.

→ One significant trend that is shaping the Android platform & the mobile app industry is the growth of progressive web apps (PWAs) & their impact on native Android app development. Here's an explanation of how this trend could influence Android app developers & businesses.

* Progressive Web Apps (PWAs)

→ PWAs are web applications that offer a native app-like experience with a web browser. They leverage modern web technologies to provide features such as offline, caches, push notifications & a responsive design. What gets PWAs apart is their ability to function seamlessly on both desktop & mobile devices.

* Impact on Android App Developers:

1. Reduced Development Effort:- Developers could build PWAs instead of traditional native Android apps, reducing the need for separate codebases & resources for different platforms. This could save time & effort.

2. Cross-platform compatibility: PWAs are designed on various platforms, including Android. Developers can reach a broader audience with a single platform, simplifying their development and maintenance processes.

3. Enhanced user experience:- PWAs can provide app-like experience to users, including offline & push notifications. Developers can focus on delivering a better user experience without having to worry about specific constraints.

* Impact on Business in the mobile App Industry

1. Cost savings: Building a PWA can be more effective than developing separate native apps for multiple platforms. This can benefit businesses by reducing development & maintenance expenses.

2. Wider reach: PWAs can be accessed through a browser, making them accessible to a broader audience without the need for app downloads. This can help businesses reach users more easily.

3. Improved engagement: The app-like features of PWAs, such as push notifications, can lead to increased user engagement.

Faster updates:- updating a PWA is often quicker and more straightforward than updating native apps through app stores.

Reduced friction:- Since PWAs don't require users to download an app, they reduce the friction associated with app installations.

- It's important to note that while APIs offer many advantages, they may not be suitable for all types of apps or businesses. Developers & business should evaluate their specific needs & target audience to determine the most suitable approach.
- Q 2 What is the purpose of LayoutInflater in android development & how does it fit into the architecture of android layout?
- LayoutInflater is used to create a new view (or layout) object from one of your XML layouts. findViewById just gives you a reference to a view that has already been created.
1. Dynamic UI creation: Android apps often need to create & display UI components dynamically based on user interactions, dates or native conditions.
2. Separation of concerns: Separating the UI layout definition (XML) from the code that manages it helps maintain a clear separation of concerns.
3. Code Reusability: Developers can reuse layout
4. Localization & Theming:
5. Efficiency
6. Custom Views

* How it fits into the Architecture

- Activity & Fragment: In Android, Activities & fragments are responsible for managing UI components.
- Binding views: After inflating a layout, developers bind views to variable in their code using "find view by ID".
- Event Handling & Interactivity: Once the layout is inflated & views are bound.
- Adaptability: The LayoutInflater allows for adaptability in UI design.

Q.3 Explain the concept of a custom Dialog Box in applications. Provide Examples to illustrate its usage.

In Android application a 'customDialog Box' often refers to a UI component that allows developers to create a customized, pop-up dialog window that can display information, collect user input or perform various actions.

* How works:-

1. Layout Definition: Developers define the layout for the custom dialog using an XML layout file.

2. Dialog creation: In code, a custom dialog is created and associated with the XML layout. You can set properties for the dialog, such as its size, style & specific behaviors.

3. Displaying the Dialog: When needed, the custom dialog can be displayed to the user.

4. User Interaction: Users can interact with the elements within the custom dialog, entering text or clicking buttons.

5. Handling action: Developers can define actions or event handlers for the UI element within the custom dialog to respond to user input.

* Examples :-

1. Login Dialog :-

- purpose :- A custom dialog for user login or authentication.
- layout :- The dialog may contain EditTexts for username and password, a "Login" button & "forgot password" link.
- use case :- Displayed when the user needs to log in or provide authentication credentials.

2. Confirmation Dialog :-

- purpose :- A dialog that takes the user for confirmation before proceeding with a critical action.
- layout :- contains a message describing the action & "yes" & "No" buttons.
- use case :- used to prevent accidental action like deleting data or making irreversible changes.

3. Custom picker Dialogs

- purpose : A dialog with a custom picker, such as a date picker, time picker or color picker.
- layout : The dialog includes the picker widget & "ok" & "cancel" buttons.

4. Information Dialog :-

- purpose: Displaying informative messages or additional layout: contains text or multimedia elements to convey
- use case: used for displaying pop-up help message, service

Q. 11 How do activities, services & the android manifest together to make an Android app. can you describe their roles & provide a basic example of how they work in a mobile app?

→ In Android app development, Activities, Services & the manifest file work together to create the structure & functionality of an app

2. Activities

- Role: Activities represent individual screens or user interface in an Android App.
- Example: Imagine a simple email app with two activities one for composing emails & another for viewing the inbox.

2. Services

- Role: Services are background components that perform running operations without a user interface.
- Example: In our email app example, a service could periodically check for new emails in the background & update the inbox.

3. Android manifest file :-

- Role: the AndroidManifest.xml file is a configuration that defines essential information about the app components, and required permissions
- Example: In the manifest file, you specify which activities & the app contains, their properties & any permission required among other things

Q.5

How does the Android manifest file import the development of an Android application? provide an example to demonstrate its significance.

The Android manifest file is a critical component in Android app development, it serves several significant purposes that impact the development & end functionality of an Android application.

* Example

1. component Declaration: The Android manifest file is where you declare all the components of your Android application, including activities, services, broadcast receivers & content providers.

→ <application>

<activity android:name = "Main Activity">

<intent-filter>

<action android:name = "android.intent.action.MAIN">

<category android:name = "android.intent.category.LAUNCHER">

</intent-filter>

</activity>

<service android:name = "MyService"/>

<receiver android:name = "MyReceiver"/>

<provider

 android:name = "MyContentProvider"

 android:authorities = "com.example.myapp.provider"/>

</application>

2. Application configuration: This includes setting the app's theme, icon, label, version information,

& defining the minimum Android API level.

```
<application  
    android:icon="@drawable/app-icon"  
    android:label="@string/app-name"  
    android:theme="@style/AppTheme"  
    android:allowBackup="true"  
    android:versionCode="1"  
    android:versionName="1.0">
```

3. permission: you declare the permissions your app to access various system resources & characteristics, location, internet.

→ <uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

4. Intent filters

5. App permission

Q4 what is the role of resources in Android development? discuss the various types of resources & their significance in creating well-structured applications provide to clarify your points

→ Resources are the additional files & static your uses, such as bitmaps layout user interface string, animation instructions & more.

* Resources type overview:

(1) Animation resources: Define pre-determined frame animations are saved res/drawable/ & accessed from R.drawable class

(2) color state list resource: Define a color resource that changes based on the view state saved res/color/ & accessed from the class

(3) Drawable resources: Define various graphics with bitmap XML

(4) Layout resources: Define the layout for your application

(5) menu resources: Define the contents of your application menu

(4) string resources:

(5) style resources : Define the look & format for UI elements

(6) font resources : Define font families & include custom fonts in XML.

(7) How does an Android service contribute to the functioning of mobile Application? Describe the process of developing an Android service

→ Define the service in the manifest: in the Android manifest.xml file, define the service with its name & intent filter
`<service android:name="my service"/>`

(8) Implement the Service's functionality: write the logic for the service inside the overridden methods

(9) Start the service : To start the service, create an Intent that specifies the service's class & call 'startService()' with the Intent.

(10) Optionally bind to the Service: You can stop

(11) Stop or unbind from service : You can stop a service using 'stopService()' or unbind from it using 'unbindService()'.

in progress