ANDROID APP BUILDING (From Scratch)

★ One-shot guide to build an App ★

Steps:-

(1) BEFORE STARTING THE APP:-

i) Define:- •) what does app do?
•) users •) features in V1? (3-5 max)
•) screens need? (Home, Details, Settings etc)

ii) Sketch UI (very rough)
Just draw boxes
•) Home → cards / List
•) Details → Title + Info
•) Add / Edit → Input + Button

⇒ This makes coding 3x easier

(2) STARTING The PROJECT:-

i) Create Project:-
•) Creat Empty Activity / Basic Activity
•) Package name: com.yourname.appname

ii) Setup App structure:-
use packages:
ui/ // Keep app clean
data/
model/
utils/

Date / /
Page No.

(3) UI (XML) – Building The Screens

Layout Rules:-

- use ConstraintLayout (safe, stable)
- use LinearLayout inside only if needed
- use RecycleView for lists
- use ScrollView for long pages
- use dp for size, sp for text, wrap_content mostly.

Basic Layout Template

code (xml)

```
<androidx.constraintlayout.widget.... >

<TextView
    android:id="@+id/title"
    android:text="Hello"
    android:textSize="22sp" />

<RecyclerView
    android:id="@+id/myList"
    android:layout_width="0dp"
    android:layout_height="0dp" />

</androidx.constraintlayout.... >
```

Date / /
Page No

(4) LISTS: LISTVIEW → ADAPTER → RECYCLER VIEW

Recyclerview Flow

1. **Model class**

```
data class User(val name: String,
                val age: Int)
```

2. **Item XML**

```
<TextView android:id="@+id/name"/>
```

3. **Adapter**

```
class UserAdapter(....): RecyclerView.
    Adapter<UserViewHolder>()
```

4. **Attach in Activity**

```
recyclerView.adapter = UserAdapter(list)
```

> RecyclerView = Box
> Adapter = Delivery boy
> ViewHolder = Single Item's container

(5) NAVIGATION (Bottom Nav, Fragments, Drawer)

Use Fragment when:

- multiple screens
- bottom navigation
- navigation drawer
- dashboard-style apps

Date / /
Page No.

Bottom Navigation setup

1. Add dependency
2. Create Fragments
3. XML → BottomNavigationView
4. Link using Navigation Components

Navigation Drawer

- comes with template
- Add menu items in menu.xml
- Handle clicks → loads fragments

(6) FRAGMENTS (imp for Apps)

basic:-

```
class HomeFragment : Fragment
        (R.layout.fragment_home)
```

Fragment lifecycle (simple memory)

- onCreateView → UI create
- onViewCreated → bindings + clicks
- onDestroyView → clean memory

When stuck, remember:-

Activities = Hosts
Fragments = Pages

Date / /
Page No

(7) WORKING WITH APIs

Retrofit Steps :-

1. Add dependency
2. Create API Interface

```
@GET("posts")
suspend fun getPosts(): List<Post>
```

3. Build Retrofit instance
4. Call inside ViewModel / Activity
5. Show data in RecyclerView

Golden Rule :-

API → Parse → Model → RecyclerView

(8) STATE MANAGEMENT

- Keep data in ViewModel
- Use LiveData or StateFlow
- Don't fetch API again on rotation

(9) Adding Features (short recipes)

- Search: • Filter list in Adapter
- Dark Mode: • Add Theme → toggle/switch
- Local DB
  - Use Room • Create entity, dao, database
- Animation:
  Use MotionLayout or simple fade

Date / /
Page No.

(10) Publishing + Polish

UI Polish!-

- Margins : 16 dp
- Corner radius : 8 dp
- Shadows : small elevation
- Use primary & secondary colors
- Keep icons consistent (Material Icons)

App Testing!-

- check all navigation paths
- Test API offline
- Landscape mode
- Small screen / large screen

Date / /
Page No

Quick Template For Common APPs:

Login Page :-

1. Email, password
2. Button
3. Validate fields
4. Navigate to HomeFragment

Notes App:-

- Add note
- RecyclerView
- Room Database
- Edit → Update
- Delete → swipe

API App:-

- Fetch list
- Loading state
- Error state
- RecyclerView

shortcuts / Tricks / Tips:-

- use ViewBinding everywhere
- Use LazyColumn when using Jetpack Compose
- Use dp/sp always
- Don't overuse nested layouts → slows UI
- commit code after every feature
- Keep your colors in colors.xml
- Always make small components (reusable)