

Họ tên: Lê Phúc Hưng

MSSV: 20215276

Mã lớp: 151902

Môn học: Phát triển ứng dụng cho thiết bị di động (IT4785)

Source Code: <https://github.com/lephuchung/HustMobile>

Bài: Lesson 7

- Use RecyclerView to display a scrollable list
- Display a list of images using cards
- Project: Dogglers app

Contents

1. Use RecyclerView to display a scrollable list	3
1.1. Before you begin	3
1.2. Creating the project	3
1.3. Setting up the list of data	3
1.3.1. Add Affirmation strings	3
1.3.2. Create a new package	4
1.3.3. Create the Affirmation data class	5
1.3.4. Create a class to be data source	6
1.3.5. Display the size of the Affirmations list in a TextView.....	7
1.4. Adding a RecyclerView to your app.....	9
1.4.1. Add a RecyclerView to the layout	9
1.4.2. Implement an Adapter for the RecyclerView	11
1.4.3. Create the Adapter	11
1.4.4. Modify the MainActivity to use a RecyclerView	14
2. Display a list of images using cards	16
2.1. Before your begin	16
2.2. Adding images to the list items	16
2.2.1. Download the images	16
2.2.2. Add support for images in the Affirmation class	16

2.2.3. Initialize list of affirmations with images	17
2.2.4. Add an ImageView to the list item layout	17
2.2.5. Update the ItemAdapter to set the image	18
2.3. Polishing the UI	20
2.3.2. Use cards.....	21
2.3.3. Change the app theme color	23
2.3.4. Change the app icon	25
3. Project: Dogglers app	26
3.1. Before you begin	26
3.2. App overview.....	26
3.3. Get Started	26
3.3.1. Download the project code.....	26
3.3.2. Open the project in Android Studio.....	26
3.3.3. Implement the layout.....	28
3.3.4. Implement the adapter	30
3.4. Test your app.....	31

1. Use RecyclerView to display a scrollable list

1.1. Before you begin

Kiến thức bạn sẽ học được

- Cách sử dụng RecyclerView để hiển thị danh sách dữ liệu.
- Cách tổ chức mã thành các gói
- Cách sử dụng trình chuyển đổi (adapter) với RecyclerView để tùy chỉnh giao diện của từng phần tử trong danh sách.

1.2. Creating the project

Tạo project Affirmations, các bước thực hiện để tạo project như mọi khi, em xin phép không viết lại ở đây, kết quả của bước tạo project như sau:

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project tree under "Affirmations". The "app" module is selected, containing "manifests", "kotlin/java", "com.example.affirmations" (which contains "MainActivity"), "res" (with "drawable", "layout" containing "activity_main.xml", "mipmap", "values", and "xml"), and "Gradle Scripts".
- MainActivity.kt:** The main code editor shows the Kotlin implementation of the `MainActivity` class. The code initializes the activity, sets its content view to "activity_main", and applies insets to handle notch support.
- activity_main.xml:** The XML layout file defines a single `ListView` element with the id `main`.
- Toolbar:** The bottom toolbar shows standard Android Studio icons for file operations, navigation, and search.
- Status Bar:** The status bar at the bottom displays system information like battery level (83%), signal strength, and the date/time (28-Oct-24).

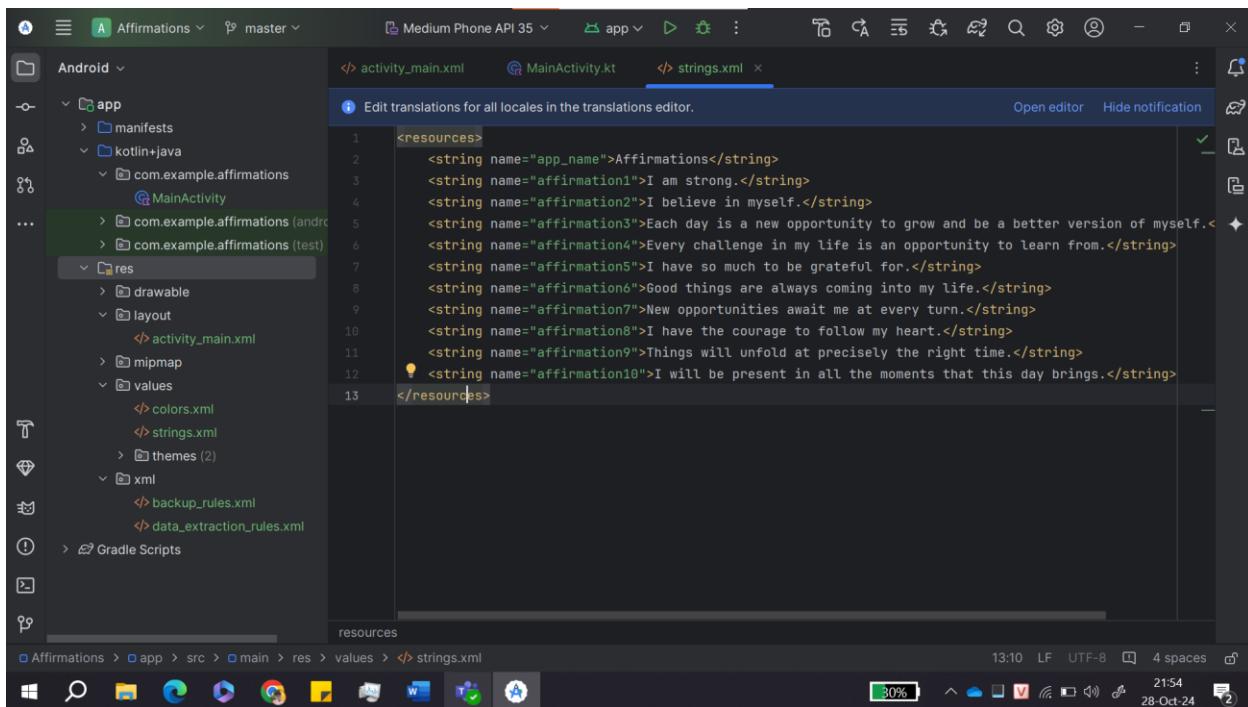
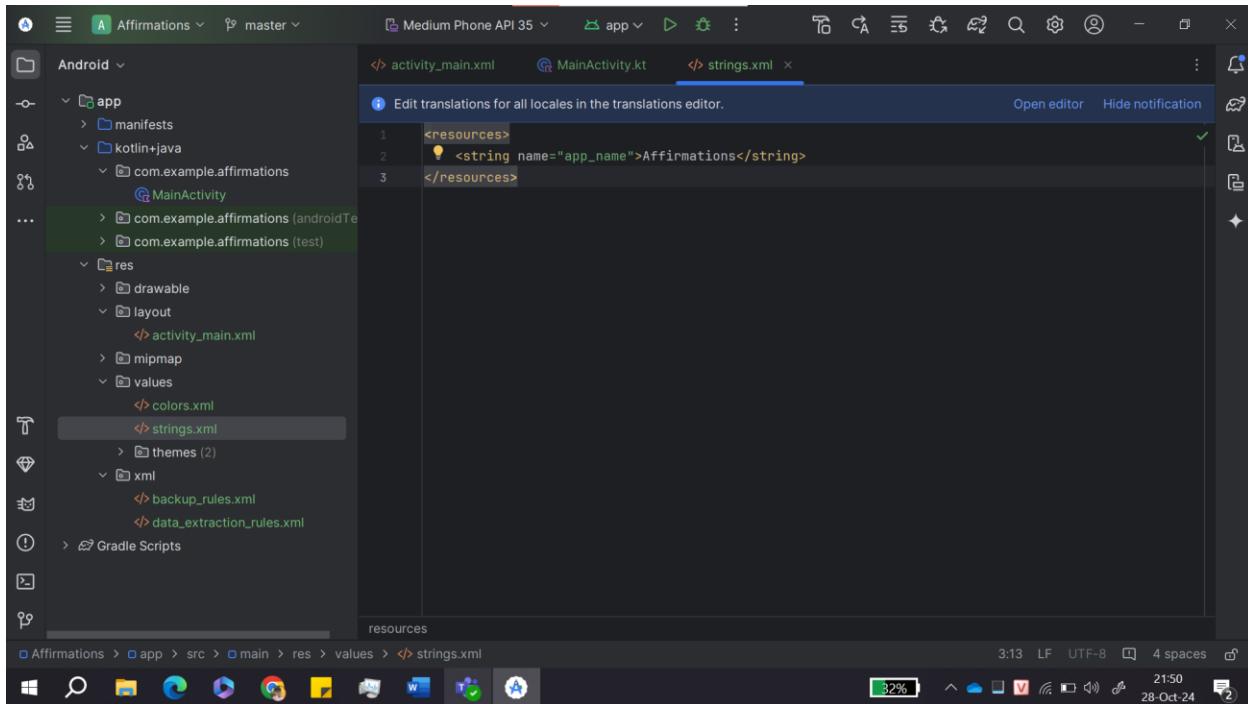
1.3. Setting up the list of data

1.3.1. Add Affirmation strings

Các bước thực hiện:

1. Mở res > values > strings.xml
2. Thêm code mẫu

Kết quả thực hiện:

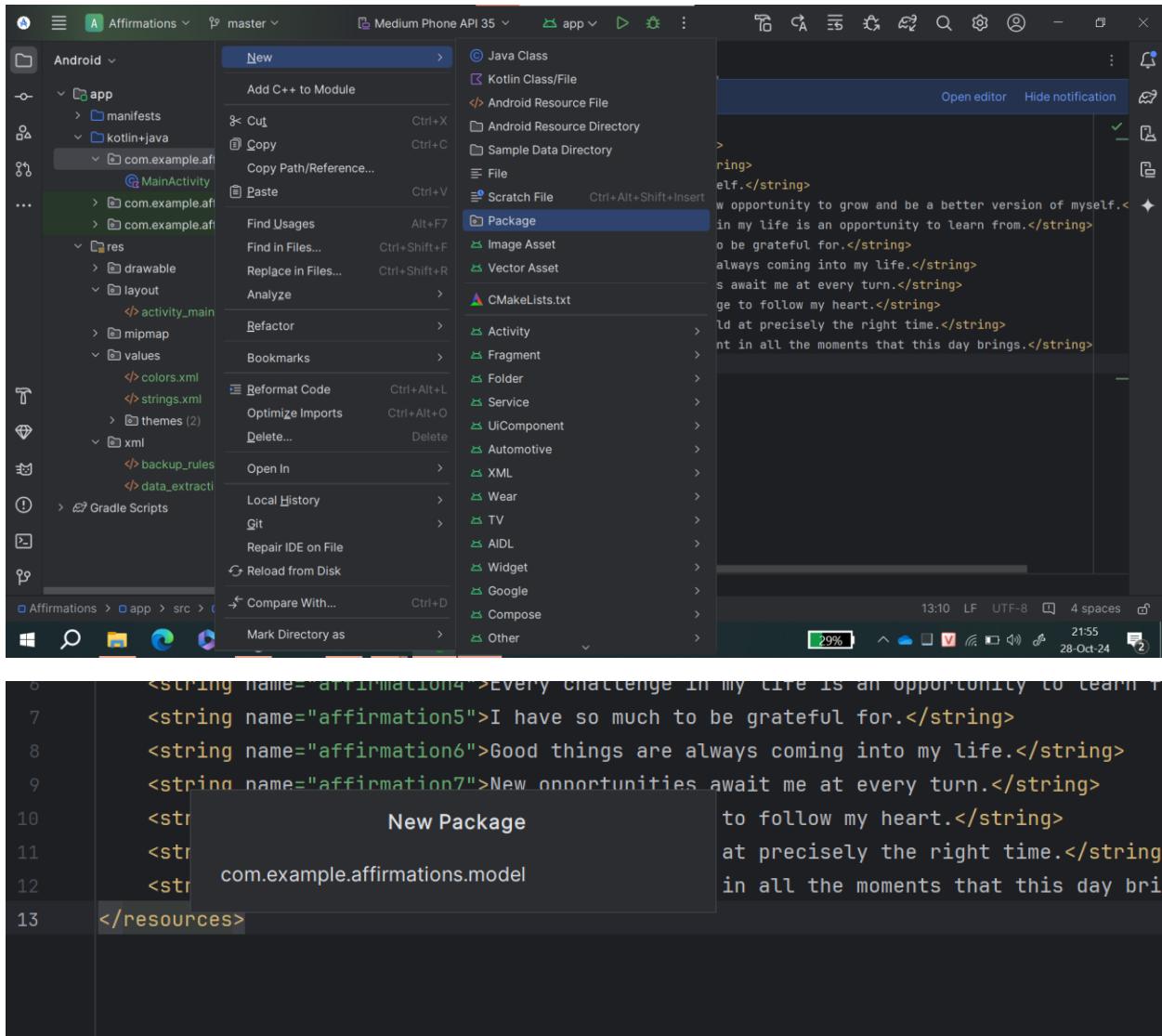


1.3.2. Create a new package

Các bước thực hiện:

1. Trong app > java > com.example.affirmations, chọn New > Package
2. Đặt tên package

Kết quả thực hiện:

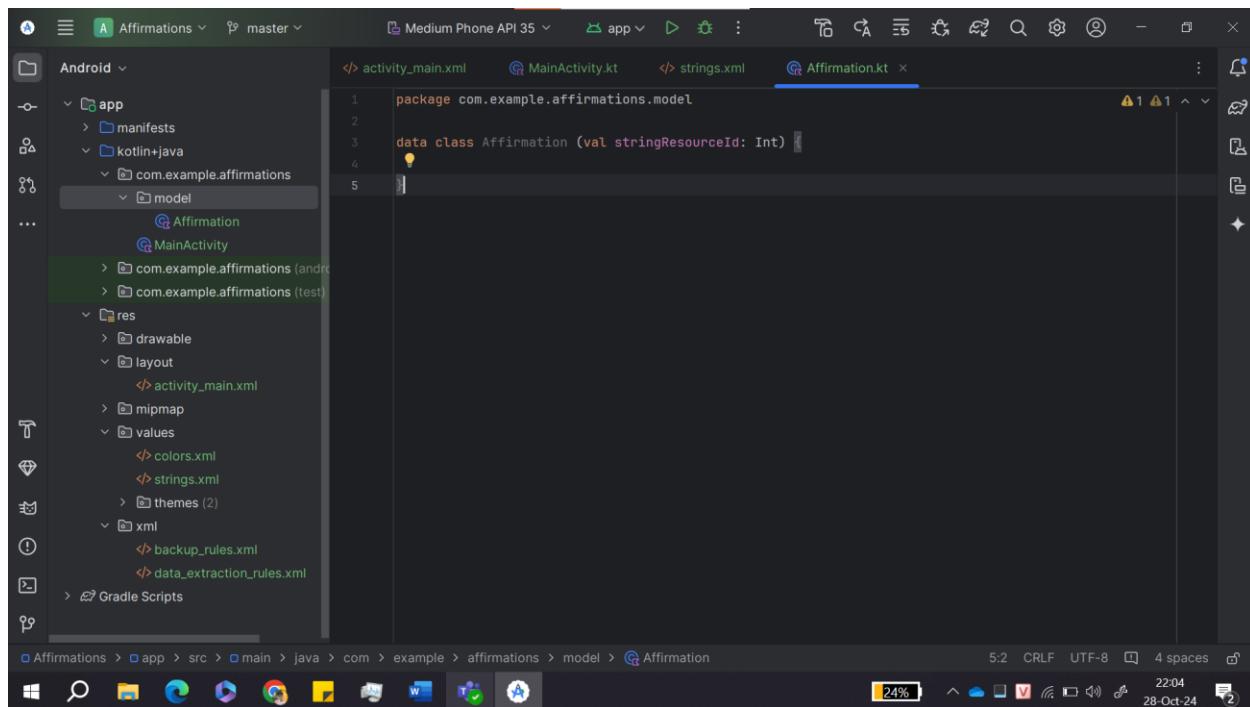
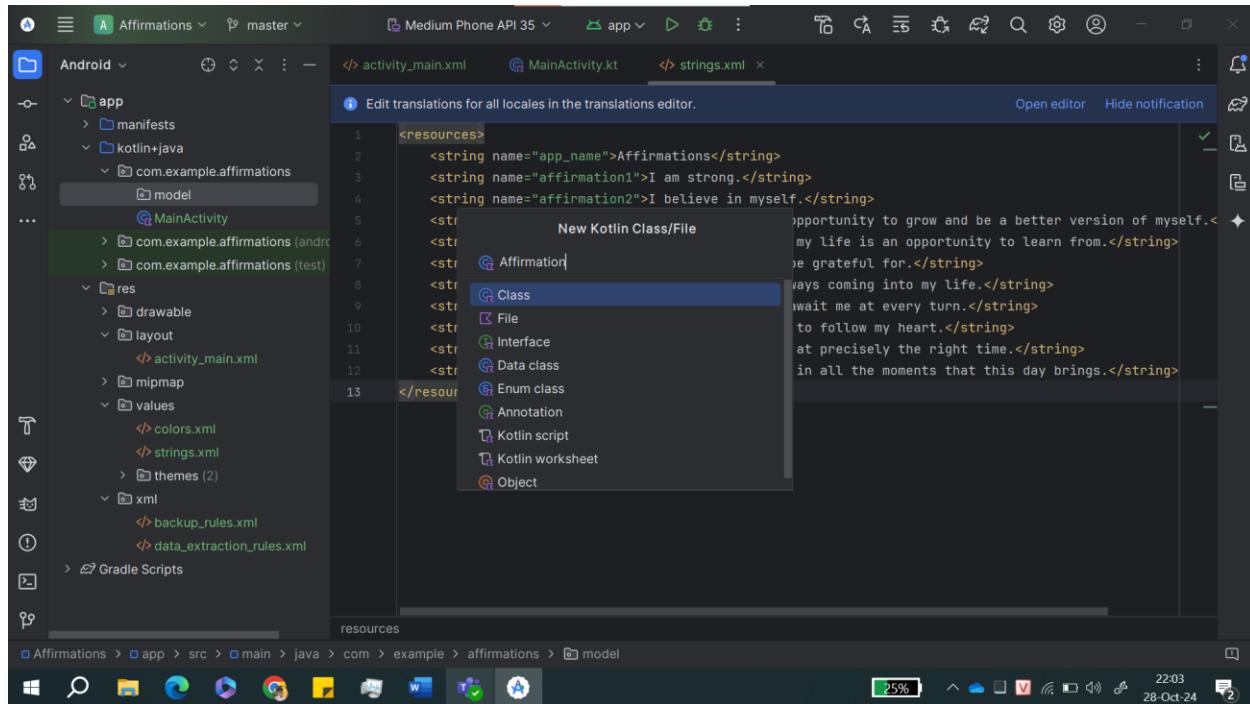


1.3.3. Create the Affirmation data class

Các bước thực hiện:

- Trong com.example.affirmations.model chọn New > Kotlin File/Class, đặt tên lớp mới là Affirmation
- Để Affirmation là một data class, thêm keyword data
- Thêm một tham số val tên là stringResourceId vào constructor

Kết quả thực hiện:



1.3.4. Create a class to be data source

Các bước thực hiện:

1. Chuột phải vào app > java > com.example.affirmations, chọn New > Package
2. Đặt tên data là phần cuối của tên gói

3. Tạo lớp Datasource cho lớp mới của gói Data
4. Trong lớp Datasource, tạo hàm mới tên loadAffirmations()
5. Đặt kiểu trả về của loadAffirmations() là List<Affirmation>
6. Thêm câu lệnh return trong loadAffirmations()
7. Sau return, gọi listOf() để tạo một List
8. Trong <>, đặt kiểu của list items là Affirmation
9. Trong ngoặc, tạo một Affirmation, truyền R.string.affirmation là resource ID
10. Thêm phần đối tượng Affirmation còn lại vào list tất cả affirmations, ngăn cách bởi dấu phẩy

```

1 package com.example.affirmations.data
2
3 import com.example.affirmations.R
4 import com.example.affirmations.model.Affirmation
5
6 class Datasource {
7     fun loadAffirmations(): List<Affirmation> {
8         return listOf<Affirmation>(
9             Affirmation(R.string.affirmation1),
10            Affirmation(R.string.affirmation2),
11            Affirmation(R.string.affirmation3),
12            Affirmation(R.string.affirmation4),
13            Affirmation(R.string.affirmation5),
14            Affirmation(R.string.affirmation6),
15            Affirmation(R.string.affirmation7),
16            Affirmation(R.string.affirmation8),
17            Affirmation(R.string.affirmation9),
18            Affirmation(R.string.affirmation10),
19        )
20    }
21 }

```

1.3.5. Display the size of the Affirmations list in a TextView

Các bước thực hiện:

1. Trong activity_main.xml, đặt tên cho id của TextView là textview
2. Trong MainActivity, trong phương thức onCreate(), thêm reference cho textview
3. Tạo một Datasource, gọi loadAffirmations(), lấy size của list trả về, chuyển nó thành chuỗi, gán nó là text của textview

Kết quả thực hiện:

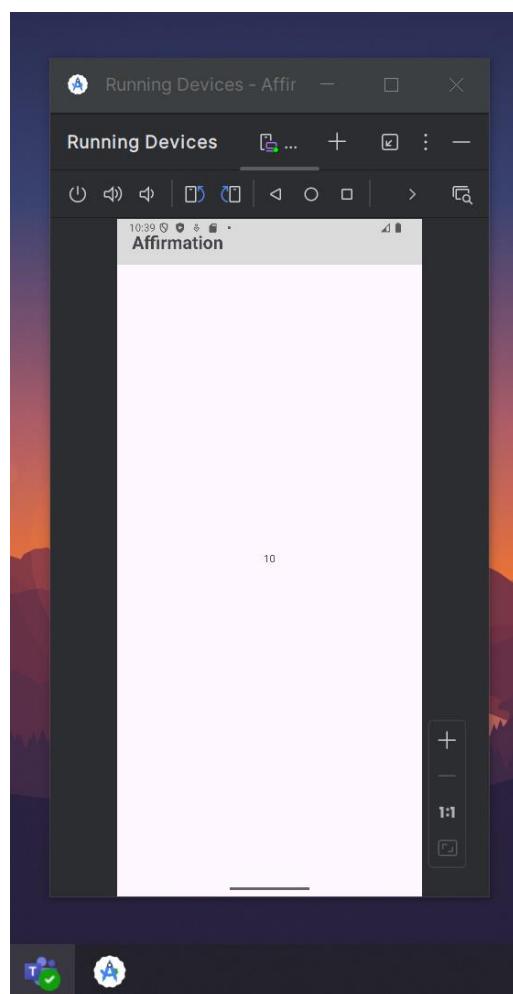
The screenshot shows the Android Studio interface. The top bar displays the project name "Affirmations", branch "master", and build configuration "Medium Phone API 35". The bottom status bar shows "Launching on devices", battery level at 32%, and the date/time "28-Oct-24".

The left sidebar shows the project structure:

- Android
- app
 - manifests
 - kotlin+java
 - com.example.affirmations
 - data
 - Datasource
 - model
 - Affirmation
 - MainActivity
 - com.example.affirmations (android)
 - com.example.affirmations (test)
 - res
 - drawable
 - layout
 - activity_main.xml
 - mipmap
 - values
 - xml
 - res (generated)
- Gradle Scripts

The main editor area shows the `MainActivity.kt` file:

```
1 package com.example.affirmations
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         enableEdgeToEdge()
9         setContentView(R.layout.activity_main)
10        val textView: TextView = findViewById(R.id.textView)
11        textView.text = Datasource().loadAffirmations().size.toString()
12    }
13 }
```



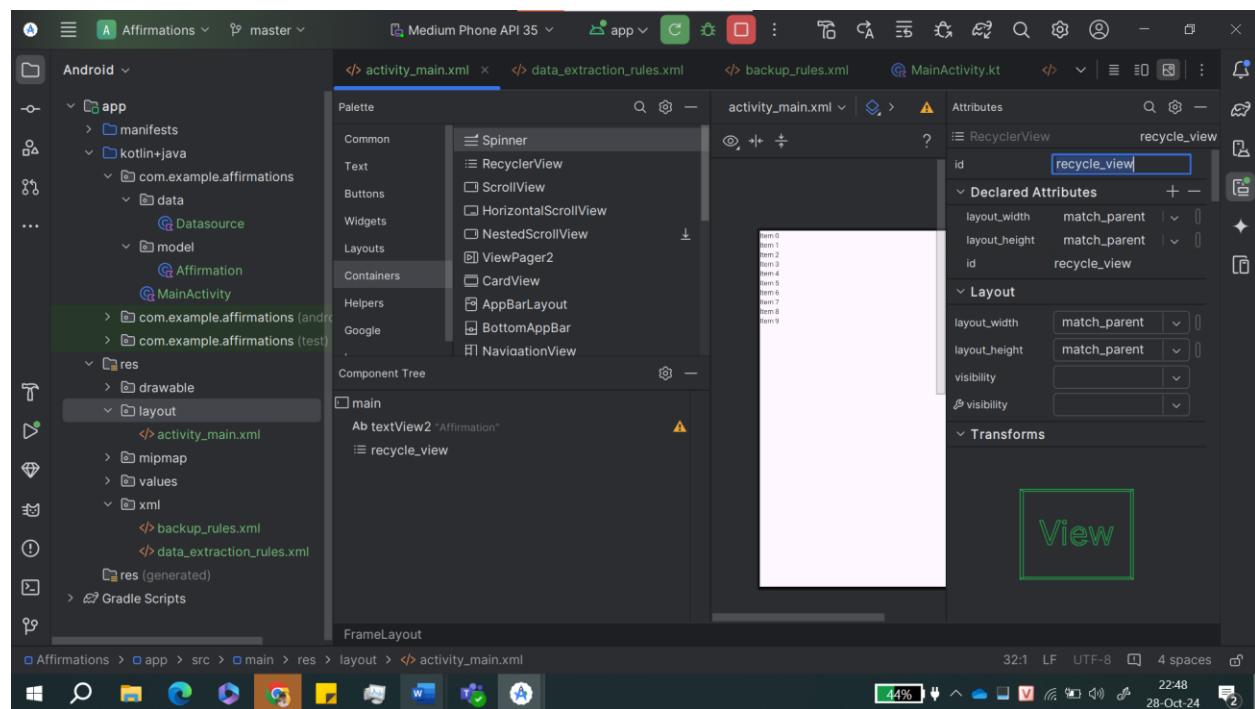
1.4. Adding a RecyclerView to your app

1.4.1. Add a RecyclerView to the layout

Các bước thực hiện:

1. Mở activity_main.xml, xóa TextView
2. Trong activity_main.xml, thay ConstraintLayout bằng FrameLayout
3. Trong Palette, chọn Containers, tìm RecyclerView, kéo một RecyclerView vào layout, đổi id thành recycle_view
4. Trong Code view, trong RecyclerView, thêm LinearLayoutManager là thuộc tính của RecyclerView
5. Trong RecyclerView, thêm android:scrollbars đặt về vertical
6. Chạy app

Kết quả thực hiện:

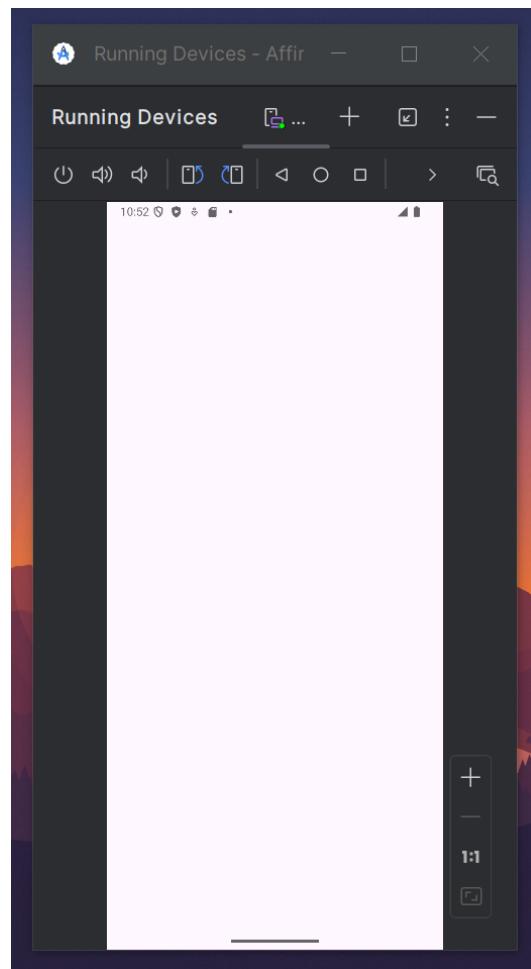


The screenshot shows the Android Studio interface. The left sidebar displays the project structure under the 'Android' tab, with the 'app' module selected. The 'layout' directory contains 'activity_main.xml'. The right panel shows the XML code for 'activity_main.xml'. The code defines a FrameLayout with a RecyclerView as a child. The bottom status bar shows the path 'Affirmations > app > src > main > res > layout > activity_main.xml'.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recycle_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scrollbars="vertical"
        app:layoutManager="LinearLayoutManager"
        />

</FrameLayout>
```



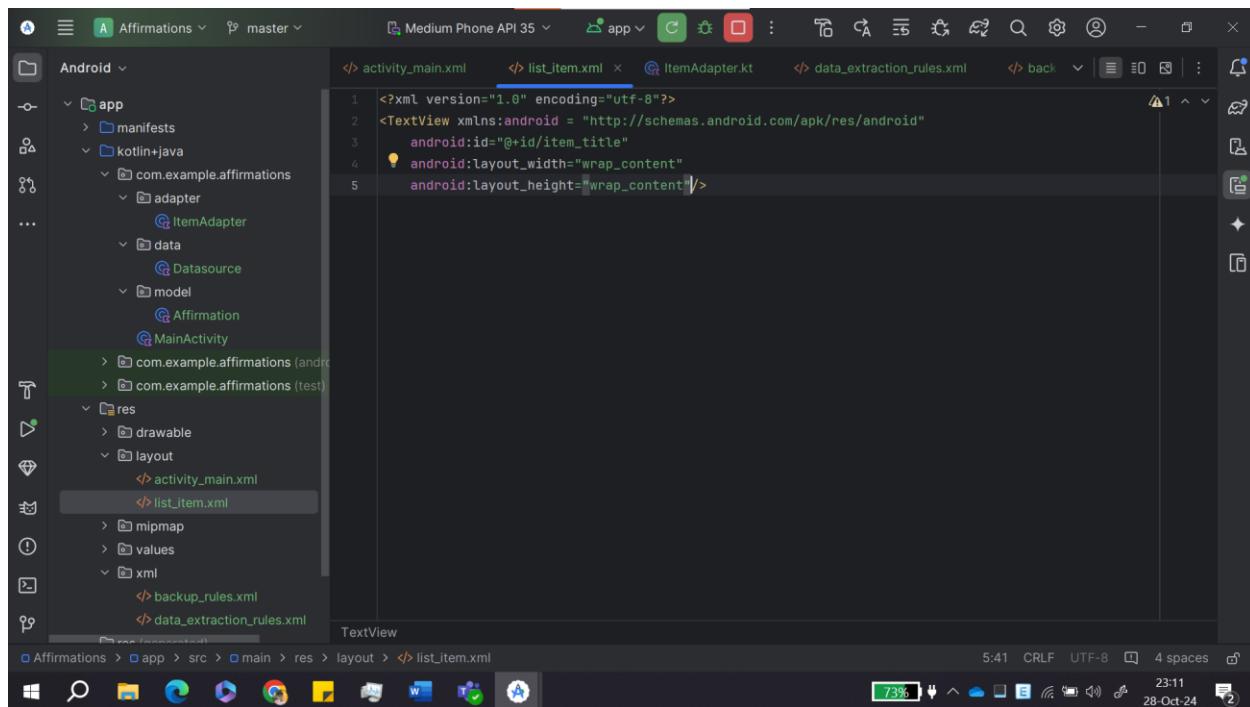
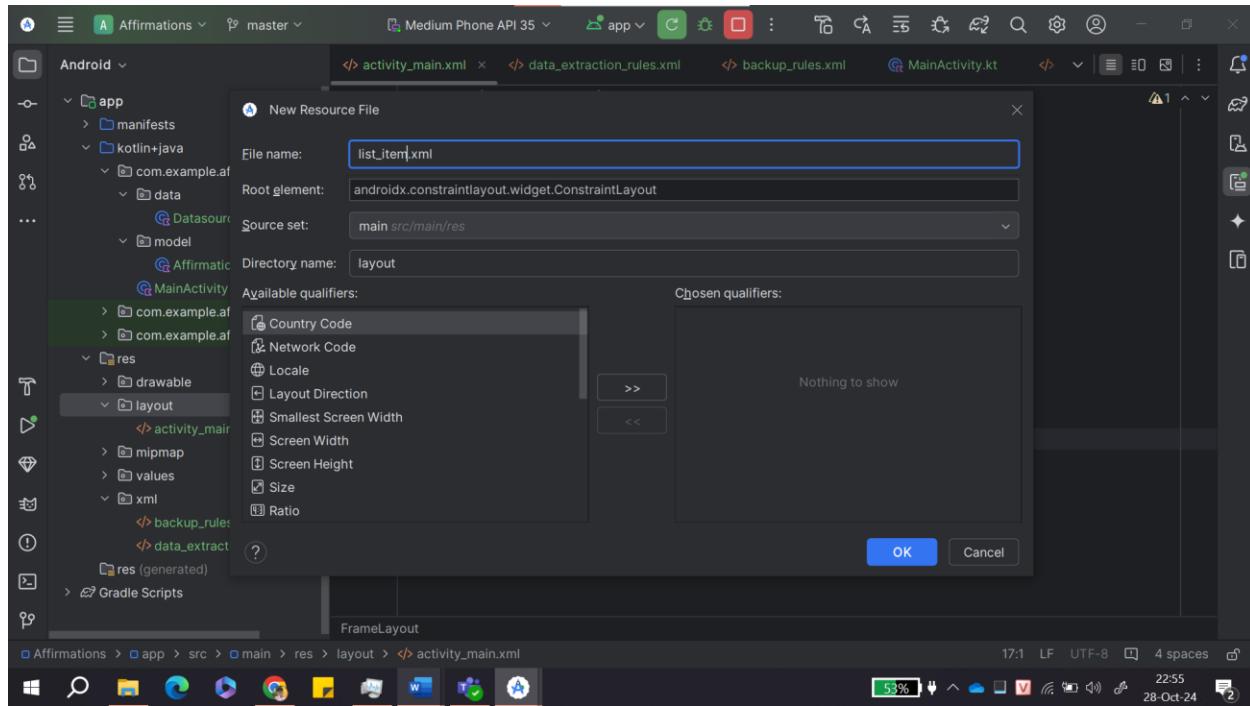
1.4.2. Implement an Adapter for the RecyclerView

1.4.3. Create the Adapter

Các bước thực hiện:

1. Trong res > layout, tạo một File trống tên list_item.xml
2. Mở list_item.xml trong Code view
3. Thêm một TextView với id item_title
4. Thêm wrap_content cho layout_width và layout_height
5. Trong app > java > com.example.affirmations và chọn New > Package
6. Package tên adapter
7. Chuột phải vào package adapter, chọn New > Kotlin File/Class
8. Đặt tên lớp là ItemAdapter
9. Thêm tham số vào constructor ItemAdapter là val dataset có kiểu là List<Affirmation>
10. Đặt dataset là private
11. Thêm tham số vào ItemAdapter là val context với kiểu Context
12. Trong lớp ItemAdapter, tạo một lớp ViewHolder
13. Thêm một private val view là tham số cho lớp ViewHolder
14. Cho ViewHolder là subclass của RecyclerView, truyền nó là tham số view vào superclass
15. Trong ViewHolder, định nghĩa một val textView có kiểu là TextView, gán nó cho view với ID item_title
16. Thêm code mẫu vào ItemAdapter
17. Đặt con trỏ chuột vào ItemAdapter và nhấn Control I, chọn cả 3 options bằng cách nhấn Shift click và nhấn OK
18. Thay getItemCount bằng code mẫu
19. Thay code của onCreateViewHolder() và onBindViewHolder() bằng code mẫu

Kết quả thực hiện:



The screenshot shows the Android Studio interface with the ItemAdapter.kt file open in the editor. The code defines a RecyclerView.Adapter named ItemAdapter that implements the ItemViewHolder interface. The adapter has a private val dataset of type List<Affirmation>. It overrides the onCreateViewHolder and onBindViewHolder methods to create a new ViewHolder and bind data respectively. The getItemCount method returns the size of the dataset.

```
package com.example.affirmations.adapter

import android.content.Context
import android.view.View
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import com.example.affirmations.R
import com.example.affirmations.model.Affirmation

class ItemAdapter(
    private val context: Context,
    private val dataset: List<Affirmation>
): RecyclerView.Adapter<ItemAdapter.ItemViewHolder>() {
    class ItemViewHolder(private val view: View) :
        RecyclerView.ViewHolder(view) {
        val textView: TextView = view.findViewById(R.id.item_title);
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ItemViewHolder {
        val itemView = LayoutInflater.from(parent.context).inflate(R.layout.list_item, parent, false)
        return ItemViewHolder(itemView)
    }

    override fun onBindViewHolder(holder: ItemViewHolder, position: Int) {
        val currentAffirmation = dataset[position]
        holder.textView.text = currentAffirmation.title
    }

    override fun getItemCount(): Int {
        return dataset.size
    }
}
```

The screenshot shows the Android Studio interface with the ItemAdapter.kt file open. A 'Implement Members' dialog is displayed over the code editor, listing methods from the RecyclerView.Adapter interface that need to be implemented. The methods listed are onCreateViewHolder, onBindViewHolder, and getItemCount.

Implement Members

- androidx.recyclerview.widget.RecyclerView.Adapter
 - onCreateViewHolder(parent: ViewGroup, viewType: Int)
 - onBindViewHolder(holder: ItemAdapter.ItemViewHolder, position: Int)
 - getItemCount(): Int

```
import com.example.affirmations.R
import com.example.affirmations.model.Affirmation

class ItemAdapter(
    private val context: Context,
    private val dataset: List<Affirmation>
): RecyclerView.Adapter<ItemAdapter.ItemViewHolder>() {
    class ItemViewHolder(private val view: View) :
        RecyclerView.ViewHolder(view) {
        val textView: TextView = view.findViewById(R.id.item_title)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ItemViewHolder {
        TODO(reason: "Not yet implemented")
    }

    override fun getItemCount(): Int {
        TODO(reason: "Not yet implemented")
    }

    override fun onBindViewHolder(holder: ItemViewHolder, position: Int) {
        TODO(reason: "Not yet implemented")
    }
}
```

```
package com.example.affirmations.adapter

import ...

class ItemAdapter(
    private val context: Context,
    private val dataset: List<Affirmation>
): RecyclerView.Adapter<ItemAdapter.ItemViewHolder>() {
    class ItemViewHolder(private val view: View) :
        RecyclerView.ViewHolder(view) {
        val textView: TextView = view.findViewById(R.id.item_title)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ItemViewHolder {
        val adapterLayout = LayoutInflater.from(parent.context)
            .inflate(R.layout.list_item, parent, attachToRoot: false)
        return ItemViewHolder(adapterLayout)
    }

    override fun getItemCount(): Int {
        return dataset.size
    }

    override fun onBindViewHolder(holder: ItemViewHolder, position: Int) {
        val item = dataset[position]
        holder.textView.text = context.resources.getString(item.stringResourceId)
    }
}
```

1.4.4. Modify the MainActivity to use a RecyclerView

Các bước thực hiện:

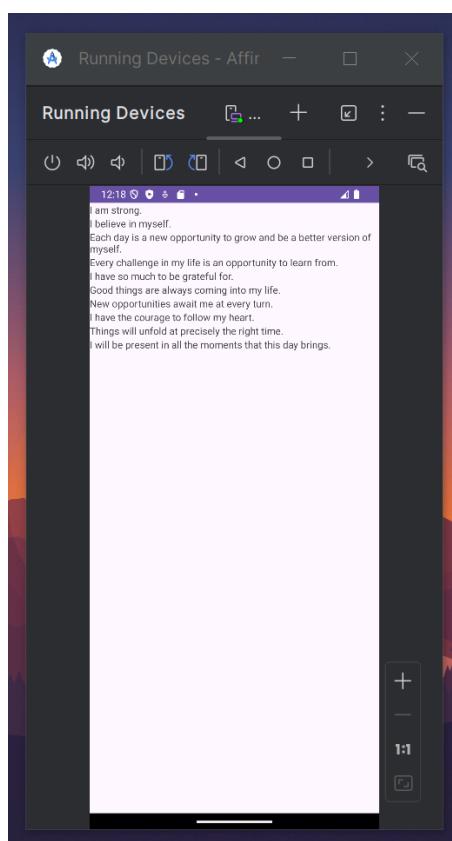
1. Mở MainActivity.kt
2. Trong phương thức onCreate(), thêm code mẫu

3. Chạy app

Kết quả thực hiện:

The screenshot shows the Android Studio interface. On the left, the Project Navigators pane displays the project structure under 'Android'. The 'app' module contains 'manifests', 'kotlin+java' (with packages like 'com.example.affirmations' containing 'Affimation', 'Adapter', 'Datasource', and 'MainActivity'), 'res' (with 'layout' containing 'activity_main.xml' and 'list_item.xml'), and 'xml' (containing 'backup_rules.xml' and 'data_extraction_rules.xml'). The right side shows the code editor for 'MainActivity.kt'. The code defines a Main Activity that overrides the onCreate method to set the content view and initialize a RecyclerView with an adapter. A message at the bottom right indicates a successful install. The bottom status bar shows the current time as 15:43 and date as 29-Oct-24.

```
package com.example.affirmations
import ...
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val myDataset = Datasource().loadAffirmations()
        val recyclerView = findViewById<RecyclerView>(R.id.recycle_view)
        recyclerView.adapter = ItemAdapter( context: this, myDataset)
        recyclerView.setHasFixedSize(true)
        val textView: TextView = findViewById(R.id.textView)
        textView.text = Datasource().loadAffirmations().size.toString()
    }
}
```



2. Display a list of images using cards

2.1. Before your begin

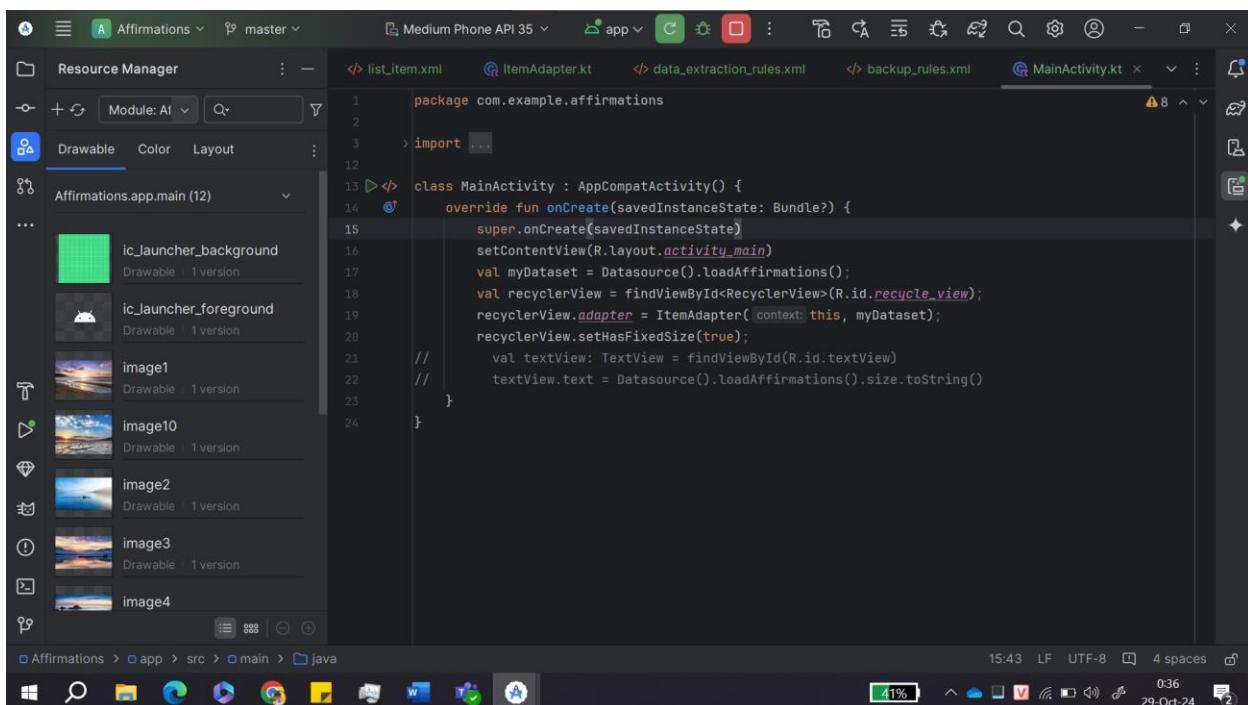
Kiến thức bạn sẽ học được:

- Cách thêm hình ảnh vào danh sách lòi tự động viên hiện trong RecyclerView.
- Cách sử dụng MaterialCardView trong bộ cục thành phần RecyclerView.
- Cách thay đổi hình ảnh giao diện người dùng để làm ứng dụng đẹp hơn.

2.2. Adding images to the list items

2.2.1. Download the images

Thêm ảnh đã tải vào folder res > drawable

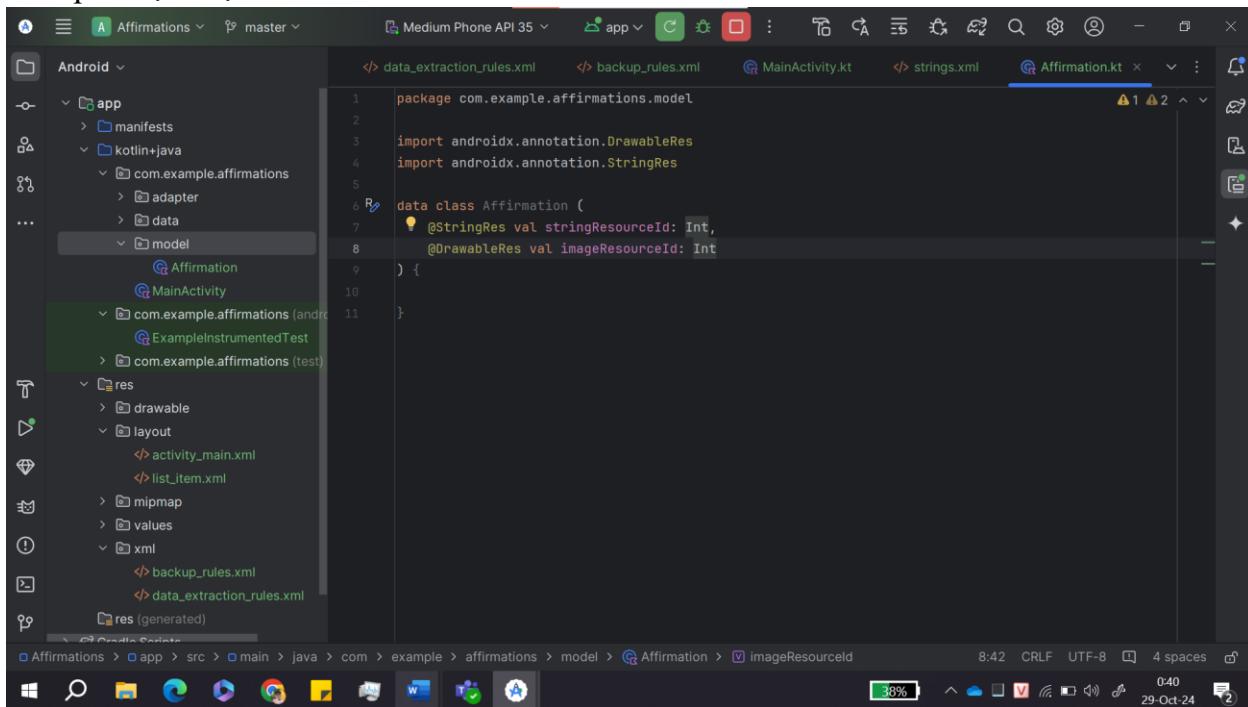


2.2.2. Add support for images in the Affirmation class

Các bước thực hiện:

1. Mở Affirmation.kt trong package model
2. Sửa theo code mẫu

Kết quả thực hiện:



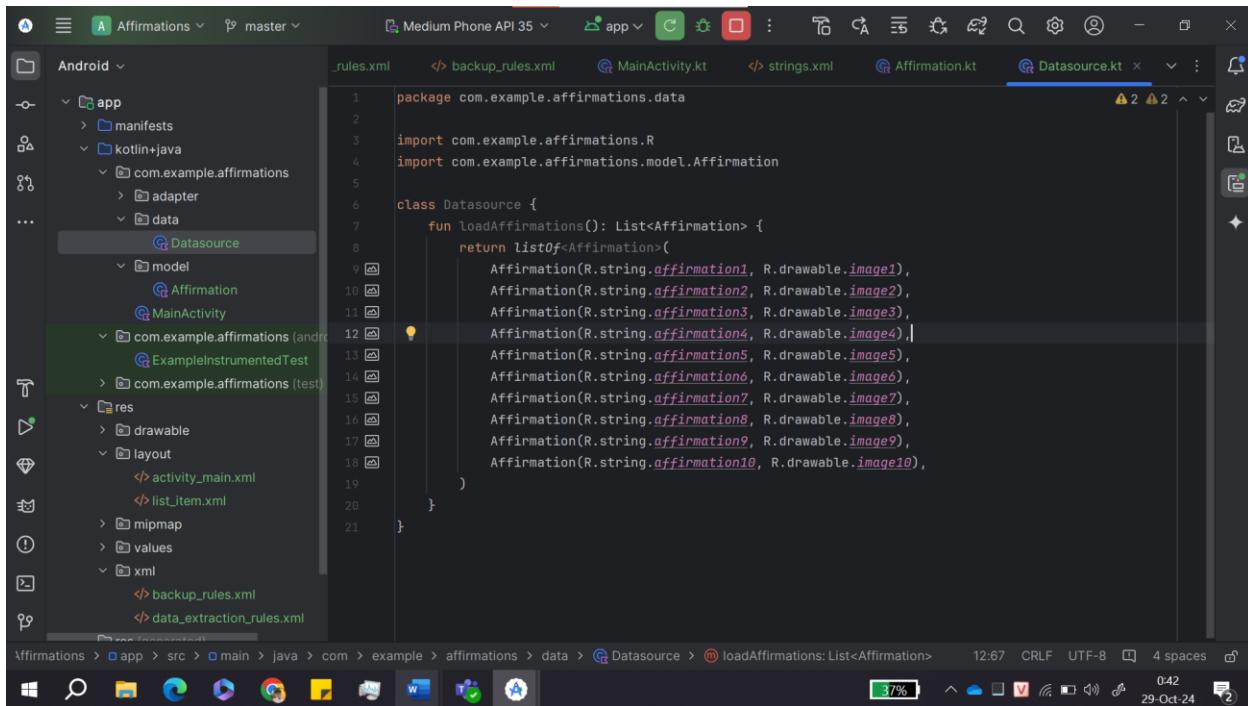
```
package com.example.affirmations.model

import androidx.annotation.DrawableRes
import androidx.annotation.StringRes

data class Affirmation (
    @StringRes val stringResourceId: Int,
    @DrawableRes val imageResourceId: Int
)
```

2.2.3. Initialize list of affirmations with images

Mở Datasource.kt, với mỗi Affirmation, thêm resource ID của một image như một argument



```
package com.example.affirmations.data

import com.example.affirmations.R
import com.example.affirmations.model.Affirmation

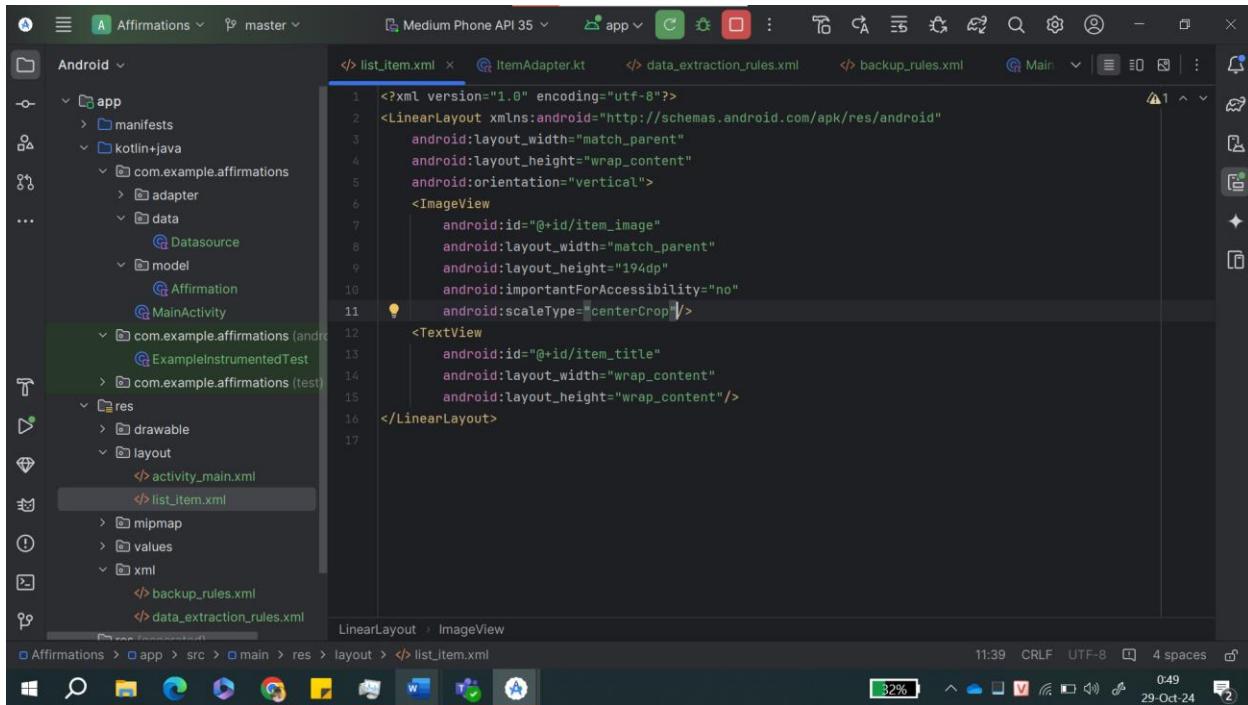
class Datasource {
    fun loadAffirmations(): List<Affirmation> {
        return listOf<Affirmation>(
            Affirmation(R.string.affirmation1, R.drawable.image1),
            Affirmation(R.string.affirmation2, R.drawable.image2),
            Affirmation(R.string.affirmation3, R.drawable.image3),
            Affirmation(R.string.affirmation4, R.drawable.image4),
            Affirmation(R.string.affirmation5, R.drawable.image5),
            Affirmation(R.string.affirmation6, R.drawable.image6),
            Affirmation(R.string.affirmation7, R.drawable.image7),
            Affirmation(R.string.affirmation8, R.drawable.image8),
            Affirmation(R.string.affirmation9, R.drawable.image9),
            Affirmation(R.string.affirmation10, R.drawable.image10)
        )
    }
}
```

2.2.4. Add an ImageView to the list item layout

Các bước thực hiện:

1. Mở res > layout > list_item.xml, thêm một LinearLayout quanh TextView có sẵn và đặt orientation thành vertical
2. Sửa theo code mẫu
3. Trong LinearLayout, trước TextView, thêm một ImageView với resource ID là item_image
4. Đặt width là match_parent, height là 194dp, đặt scaleType là centerCrop, đặt importantForAccessibility là no

Kết quả thực hiện:



The screenshot shows the Android Studio interface with the code editor open to the file `list_item.xml`. The code defines a linear layout with an image view and a text view.

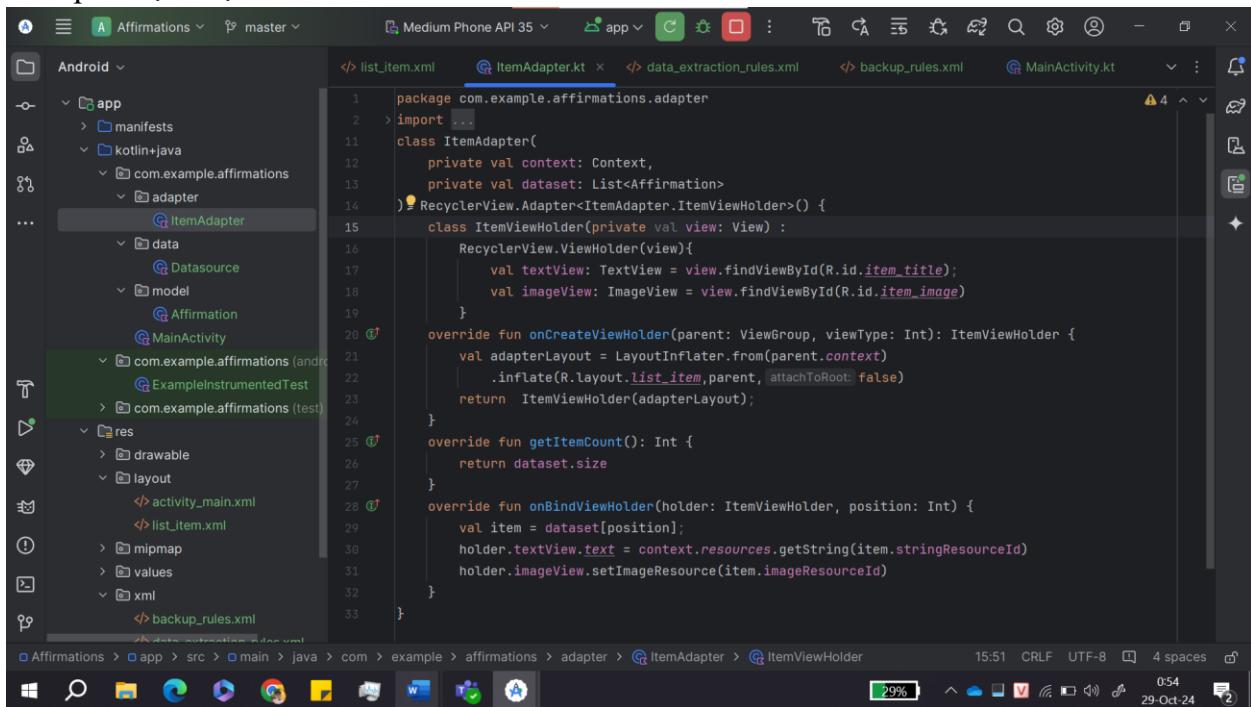
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <ImageView
        android:id="@+id/item_image"
        android:layout_width="match_parent"
        android:layout_height="194dp"
        android:importantForAccessibility="no"
        android:scaleType="centerCrop"/>
    <TextView
        android:id="@+id/item_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

2.2.5. Update the ItemAdapter to set the image

Các bước thực hiện:

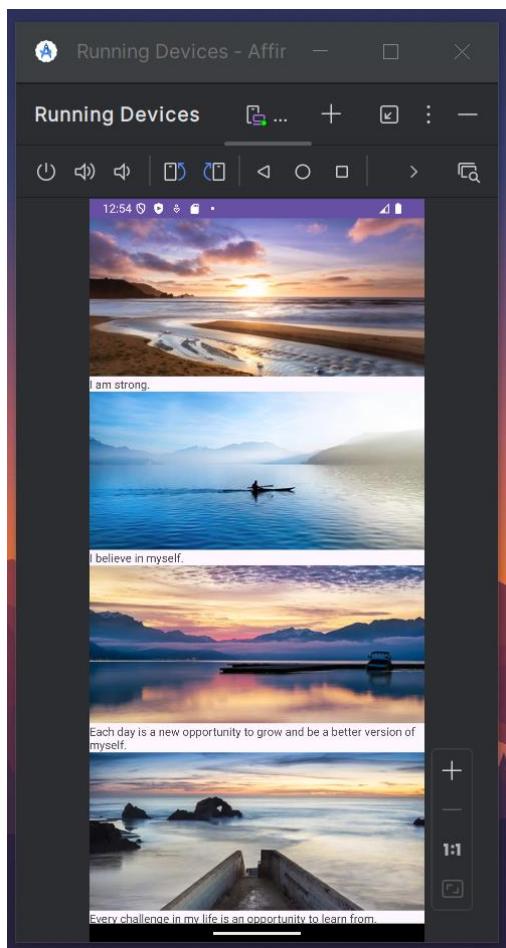
1. Mở ItemAdapter.kt, tới ItemViewHolder class
2. Sửa code theo mẫu
3. Chạy app

Kết quả thực hiện:



The screenshot shows the Android Studio interface with the code editor open to the `ItemAdapter.kt` file. The code defines a RecyclerView.Adapter for displaying affirmations. It includes methods for onCreateViewHolder, getItemCount, and onBindViewHolder, along with ViewHolder and Adapter classes.

```
1 package com.example.affirmations.adapter
2 import ...
3 class ItemAdapter(
4     private val context: Context,
5     private val dataset: List<Affirmation>
6 ) : RecyclerView.Adapter<ItemAdapter.ItemViewHolder>() {
7     class ItemViewHolder(private val view: View) :
8         RecyclerView.ViewHolder(view) {
9         val textView: TextView = view.findViewById(R.id.item_title);
10        val imageView: ImageView = view.findViewById(R.id.item_image)
11    }
12    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ItemViewHolder {
13        val adapterLayout = LayoutInflater.from(parent.context)
14            .inflate(R.layout.list_item, parent, false)
15        return ItemViewHolder(adapterLayout)
16    }
17    override fun getItemCount(): Int {
18        return dataset.size
19    }
20    override fun onBindViewHolder(holder: ItemViewHolder, position: Int) {
21        val item = dataset[position];
22        holder.textView.text = context.resources.getString(item.stringResourceId)
23        holder.imageView.setImageResource(item.imageResourceId)
24    }
25 }
```



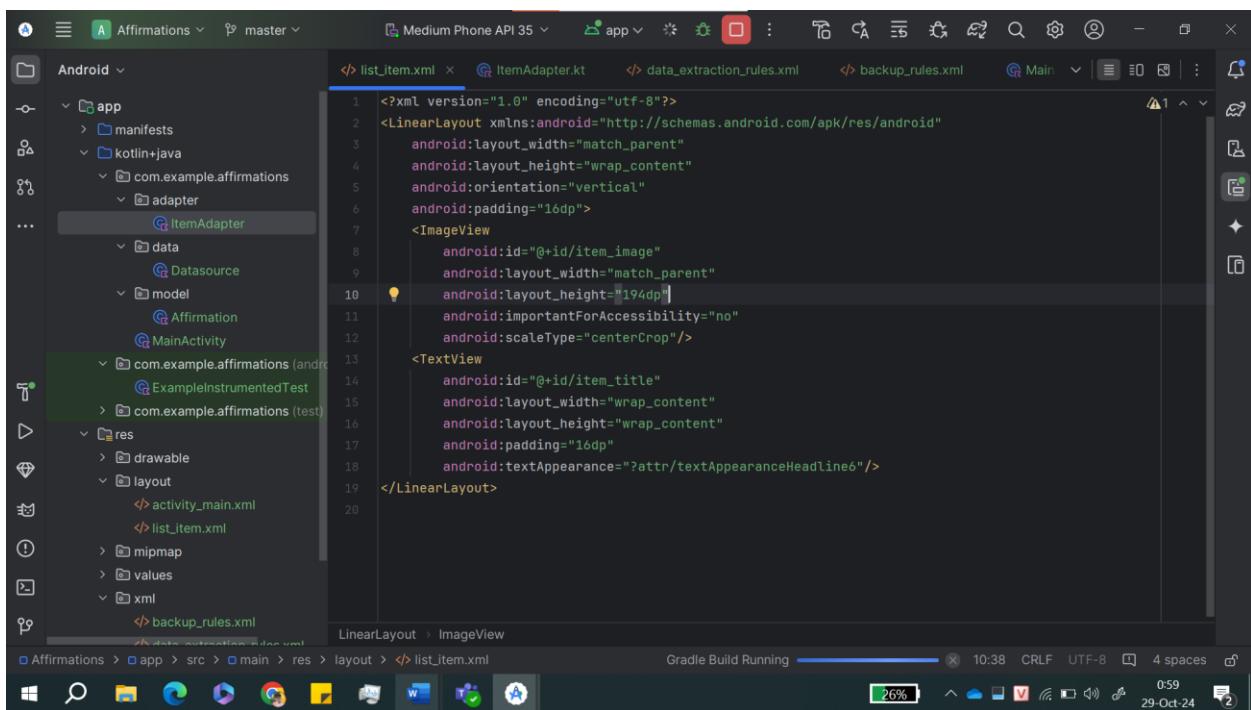
2.3. Polishing the UI

2.3.1. Add padding

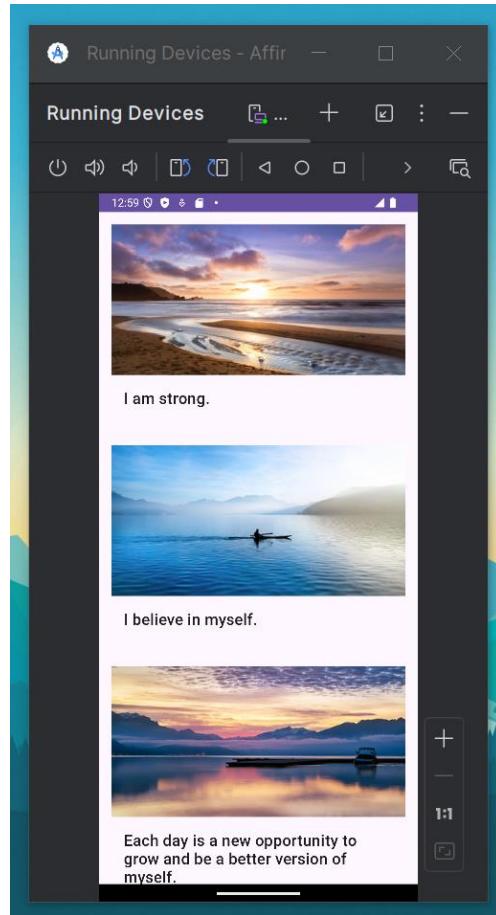
Các bước thực hiện:

1. Mở item_list.xml
2. Thêm padding 16dp vào LinearLayout
3. Thay đổi TextView theo mẫu
4. Chạy app

Kết quả thực hiện:



```
<?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="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">
    <ImageView
        android:id="@+id/item_image"
        android:layout_width="match_parent"
        android:layout_height="194dp"
        android:importantForAccessibility="no"
        android:scaleType="centerCrop"/>
    <TextView
        android:id="@+id/item_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="16dp"
        android:textAppearance="?attr/textAppearanceHeadline6"/>
</LinearLayout>
```



2.3.2. Use cards

Các bước thực hiện:

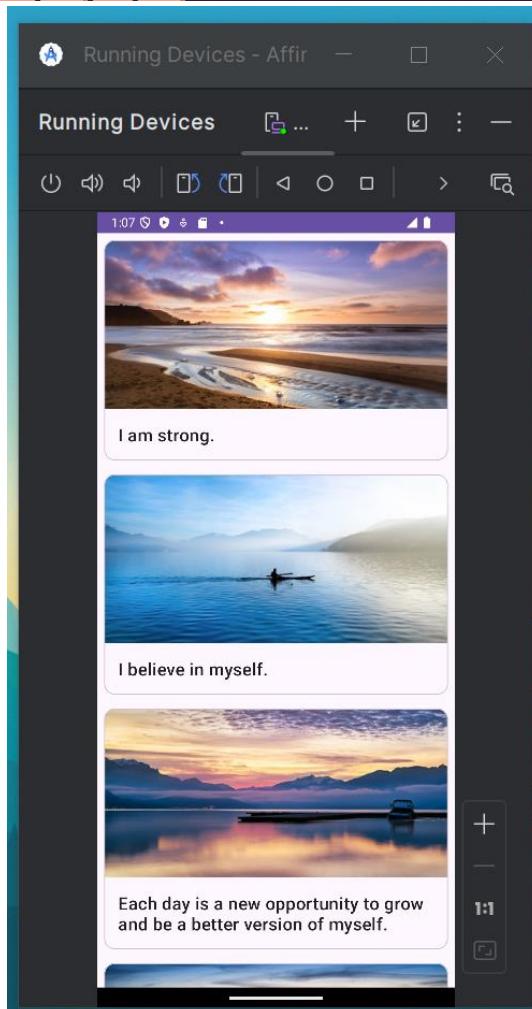
1. Sửa code theo mẫu
2. Chạy app

Kết quả thực hiện:

The screenshot shows the Android Studio interface. The left sidebar displays the project structure under the 'Android' tab, including the app module with its sub-directories like manifests, kotlin+java, and layout. The 'layout' directory contains files such as activity_main.xml and list_item.xml. The main editor window shows the XML code for list_item.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.material.card.MaterialCardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <ImageView
            android:id="@+id/item_image"
            android:layout_width="match_parent"
            android:layout_height="194dp"
            android:importantForAccessibility="no"
            android:scaleType="centerCrop"/>
        <TextView
            android:id="@+id/item_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:padding="16dp"
            android:textAppearance="?attr/textAppearanceHeadline6"/>
    </LinearLayout>
</com.google.android.material.card.MaterialCardView>
```

A status bar at the bottom indicates the build was successful: "Install successfully finished in 1s 477 ms." The bottom right corner shows the date and time: "11:19 29-Oct-24".

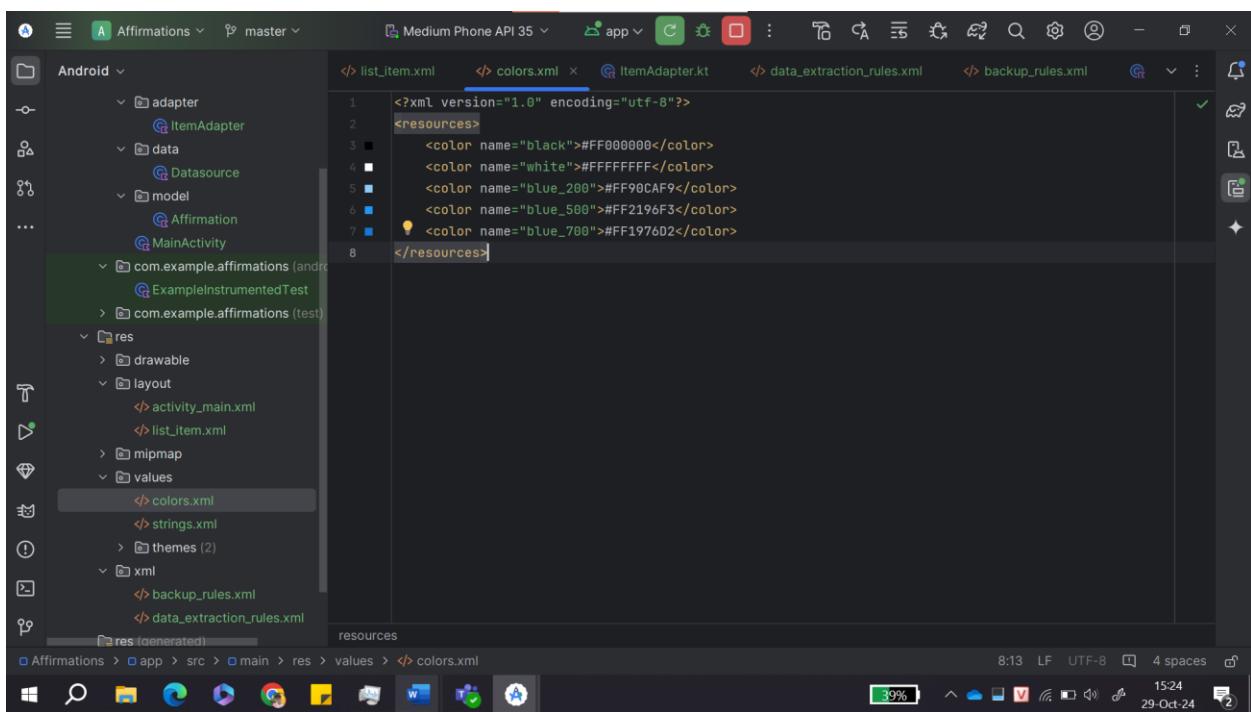


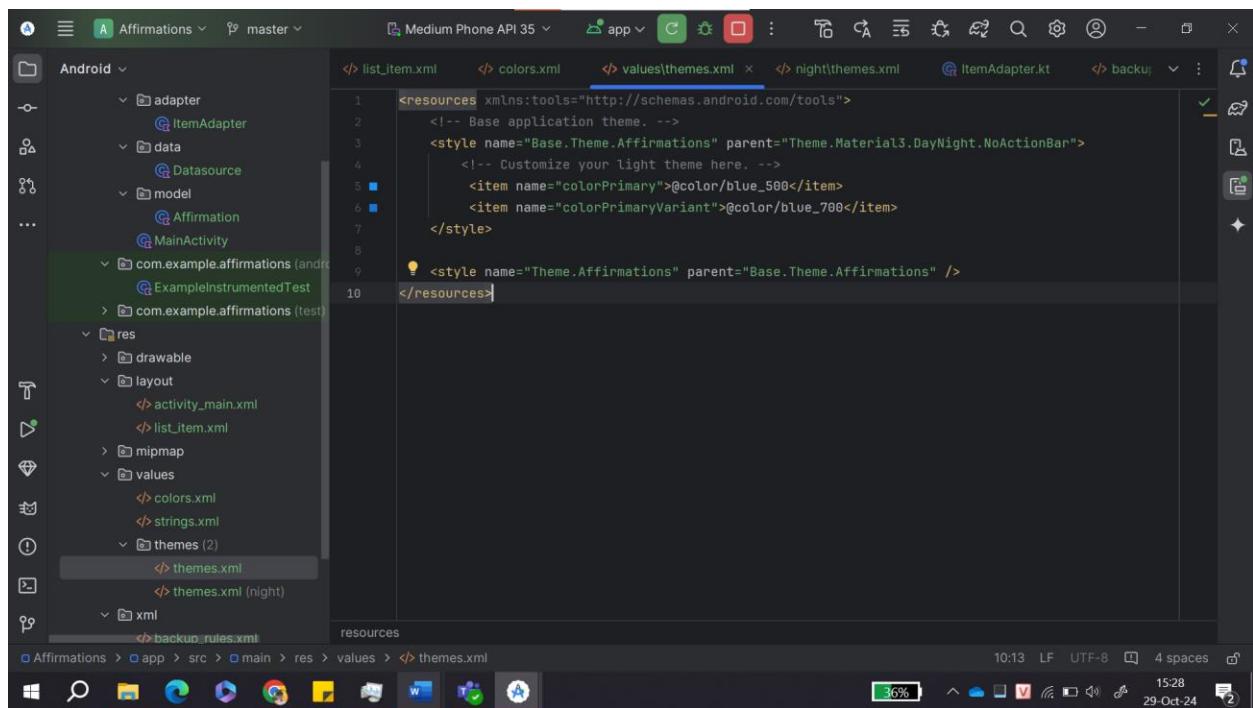
2.3.3. Change the app theme color

Các bước thực hiện:

1. Mở res > values > colors.xml
2. Thêm màu
3. Mở res > values > themes > themes.xml
4. Tìm mục <!-- Primary brand color. -->
5. Thêm colorPrimary để dùng @color/blue_500
6. Tương tự với colorPrimaryVariant thành @color/blue_700
7. Chạy app

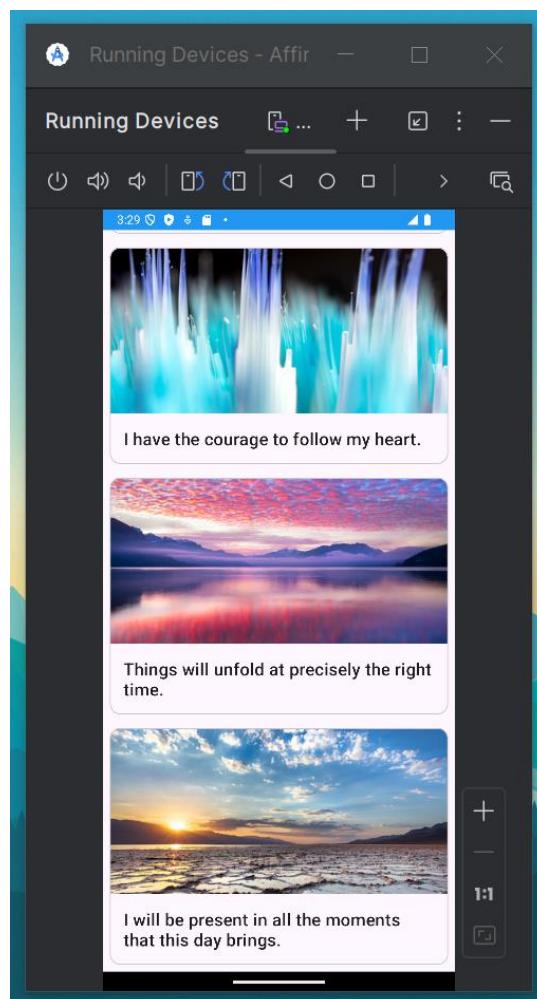
Kết quả thực hiện:





The screenshot shows the Android Studio interface with the project navigation bar at the top. The left sidebar displays the project structure under the 'Android' tab, including packages like com.example.affirmations and its sub-components: adapter, data, model, and Main Activity. The 'values' folder is expanded, showing files like colors.xml, strings.xml, and themes.xml. The 'themes.xml' file is currently selected and open in the main editor area. The code in the editor is as follows:

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.Affirmations" parent="Theme.Material3.DayNight.NoActionBar">
        <!-- Customize your light theme here. -->
        <item name="colorPrimary">@color/blue_500</item>
        <item name="colorPrimaryVariant">@color/blue_700</item>
    </style>
    <!-- Theme Affirmations -->
    <style name="Theme.Affirmations" parent="Base.Theme.Affirmations" />
</resources>
```

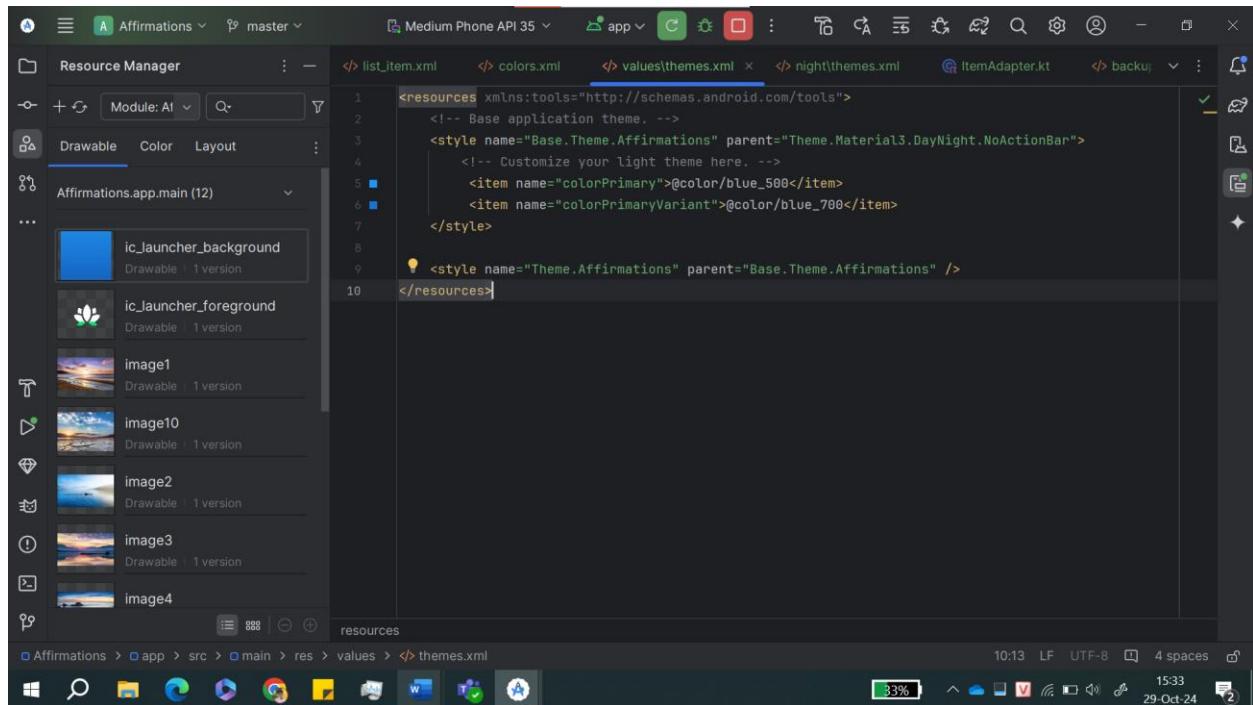


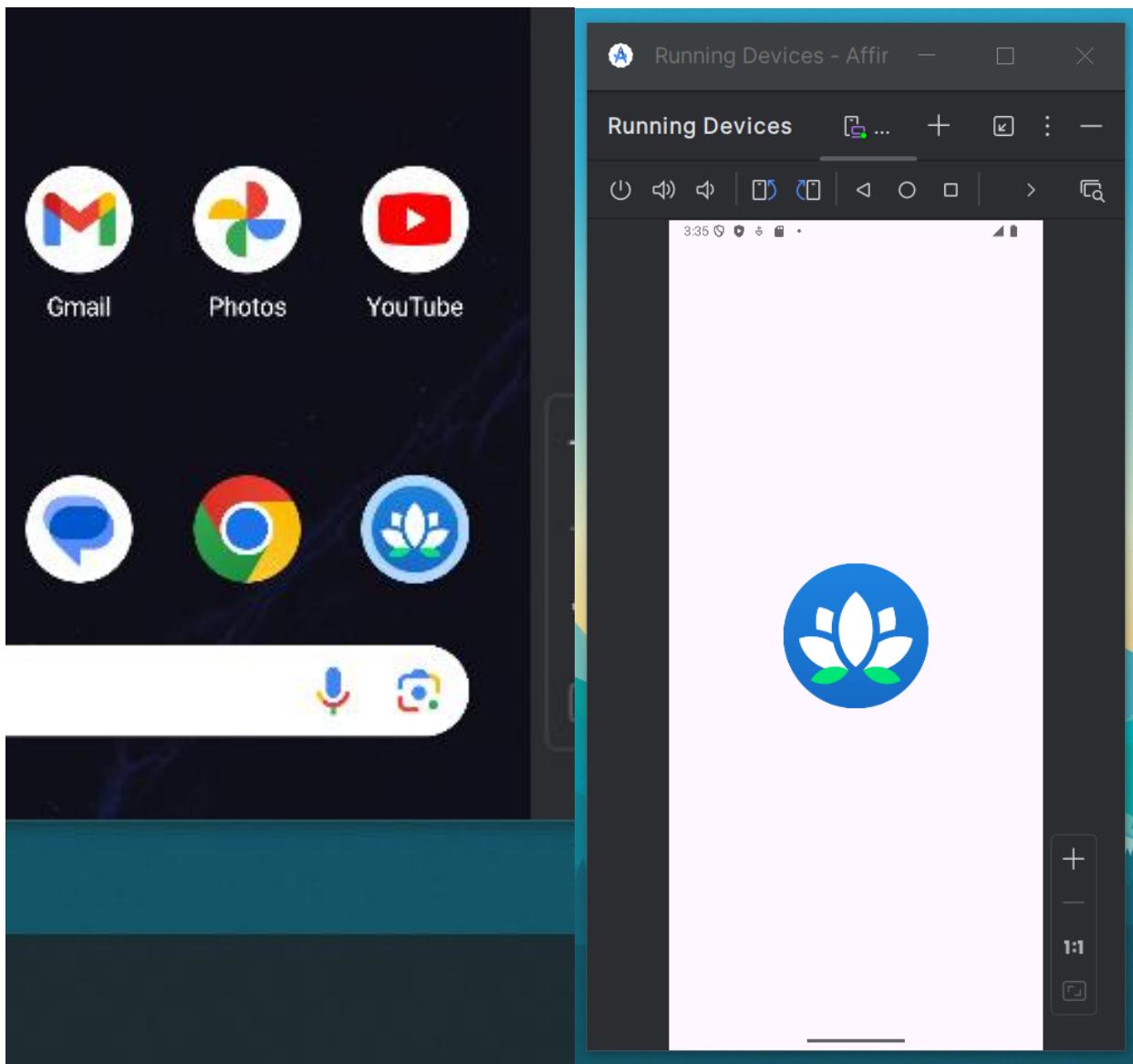
2.3.4. Change the app icon

Các bước thực hiện:

1. Tải file về
2. Xóa 2 file drawable/ic_launcher_background.xml và drawable-v24/ic_launcher_foreground.xml
3. Chuột phải res > drawable, chọn New > Image Asset
4. Trong Configure Image Asset, chọn Foreground layer
5. Tìm nhãn Path, chọn icon trong Path textbox
6. Tìm và mở ic_launcher_foreground.xml đã tải về
7. Tương tự với background
8. Chọn Next, chọn Finish
9. Chạy app

Kết quả thực hiện:





3. Project: Dogglers app

3.1. Before you begin

3.2. App overview

3.3. Get Started

3.3.1. Download the project code

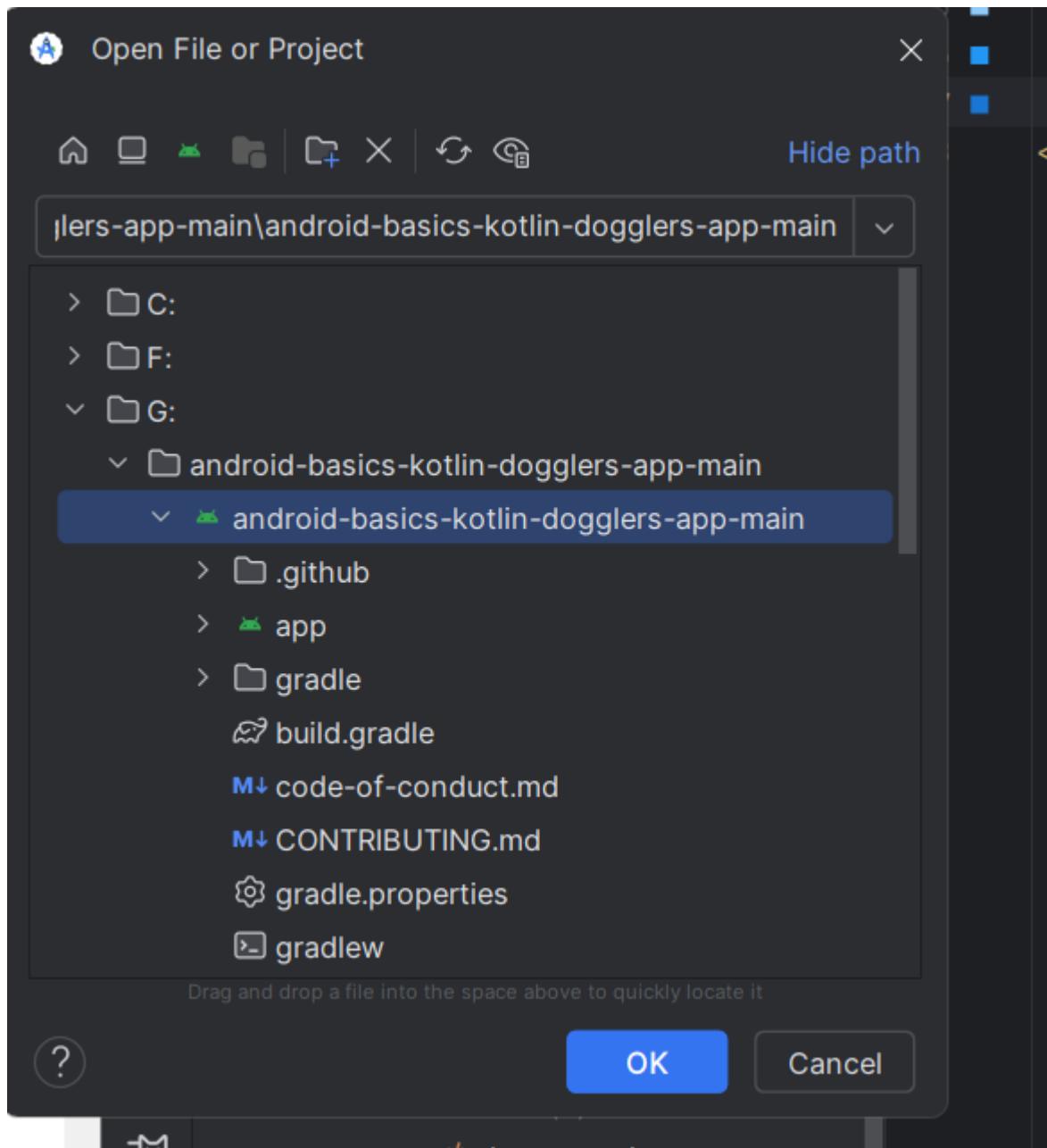
	android-basics-kotlin-dogglers-app-main.zip	10/28/2024 11:15 PM	Compressed (zipped)...	1,024 KB
--	---	---------------------	------------------------	----------

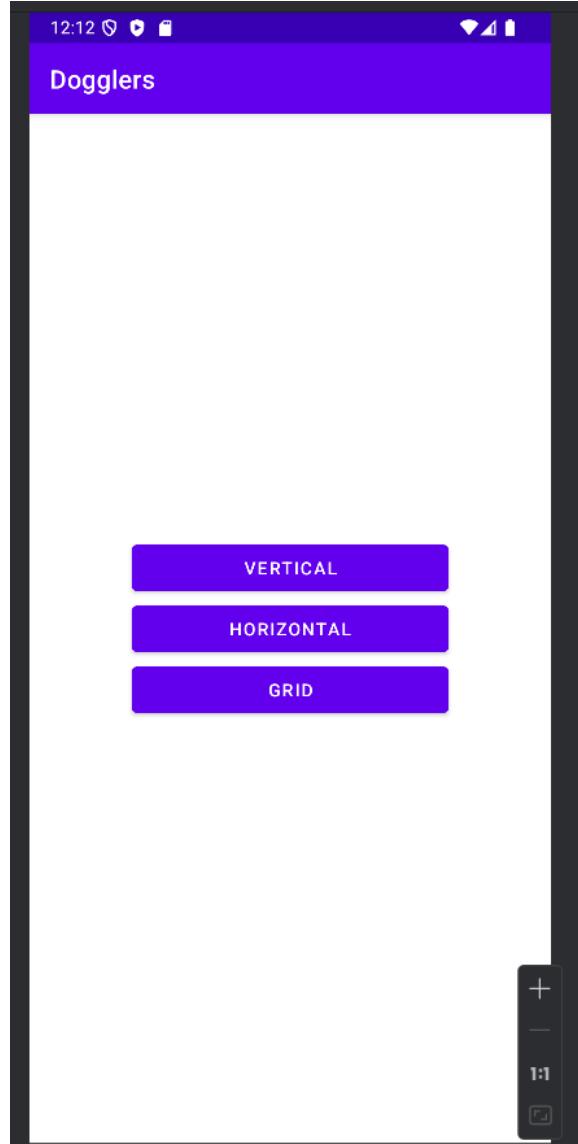
3.3.2. Open the project in Android Studio

Các bước thực hiện:

1. Mở project
2. Chạy app

Kết quả thực hiện:





3.3.3. Implement the layout

Các bước thực hiện:

1. Build the layout for vertical and horizontal lists
2. Build the grid layout

Kết quả thực hiện:

```
<TextView
    android:id="@+id/age_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="16dp"
    android:layout_marginBottom="8dp"
    android:textAppearance="@style/TextAppearance.MaterialComponents.Body1"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/name_text"
    tools:text="Age: 7" />

<TextView
    android:id="@+id/hobbies_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:textAppearance="@style/TextAppearance.MaterialComponents.Body1"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    tools:text="Hobbies: sunbathing" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<ImageView
    android:id="@+id/image_view"
    android:layout_width="match_parent"
    android:layout_height="194dp"
    android:scaleType="centerCrop"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/name_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:textAppearance="@style/TextAppearance.MaterialComponents.Headline6"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/image_view"
    tools:text="Tzeitel" />

<TextView
    android:id="@+id/age_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
<TextView
    android:id="@+id/age_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="16dp"
    android:layout_marginBottom="8dp"
    android:textAppearance="@style/TextAppearance.MaterialComponents.Body1"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/name_text"
    tools:text="Age: 7" />

<TextView
    android:id="@+id/hobbies_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:textAppearance="@style/TextAppearance.MaterialComponents.Body1"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    tools:text="Hobbies: sunbathing" />
/>androidx.constraintlayout.widget.ConstraintLayout>
<!-- .google.android.material.card.MaterialCardView -->
```

3.3.4. Implement the adapter

```

/*
Copyright (C) 2021 The Android Open Source Project. ...
*/
package com.example.doggles.adapter

import ...

/**
 * Adapter to inflate the appropriate list item layout and populate the view with information
 * from the appropriate data source
 */
class DogCardAdapter(
    private val context: Context?,
    private val layout: Int
) : RecyclerView.Adapter<DogCardAdapter.DogCardViewHolder>() {

    val dataset: List<Dog> = DataSource.dogs

    class DogCardViewHolder(view: View?) : RecyclerView.ViewHolder(view!!) {
        val dog_image: ImageView? = view?.findViewById(R.id.image_view)
        val dog_name: TextView? = view?.findViewById(R.id.name_text)
        val dog_age: TextView? = view?.findViewById(R.id.age_text)
        val dog_hobbies: TextView? = view?.findViewById(R.id.hobbies_text)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): DogCardViewHolder {
        val adapterLayout = when (layout) {
            3 -> LayoutInflater.from(parent.context)
                .inflate(R.layout.grid_list_item, parent, attachToRoot: false)

            else -> LayoutInflater.from(parent.context)
                .inflate(R.layout.vertical_horizontal_list_item, parent, attachToRoot: false)
        }
        return DogCardViewHolder(adapterLayout)
    }

}

```

```

override fun getItemCount(): Int = dataset.size

override fun onBindViewHolder(holder: DogCardViewHolder, position: Int) {
    val dog = dataset[position]
    val resources = context?.resources
    holder.dog_image?.setImageResource(dog.imageResourceId)
    holder.dog_name?.text = dog.name
    holder.dog_age?.text = dog.age
    holder.dog_hobbies?.text = "Hobbies: ${dog.hobbies}"
}

```

3.4. Test your app

Khi chạy phần test button ở dưới ta được:

