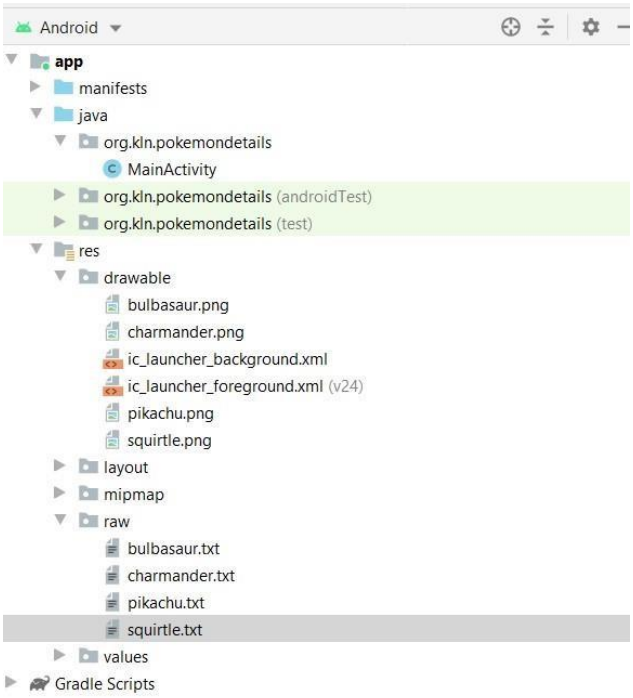
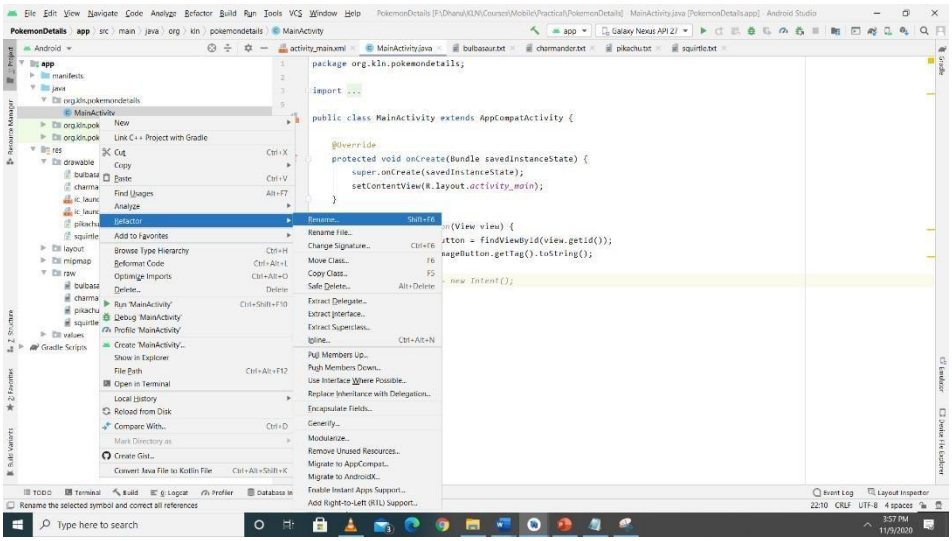
Practice Guide - 02

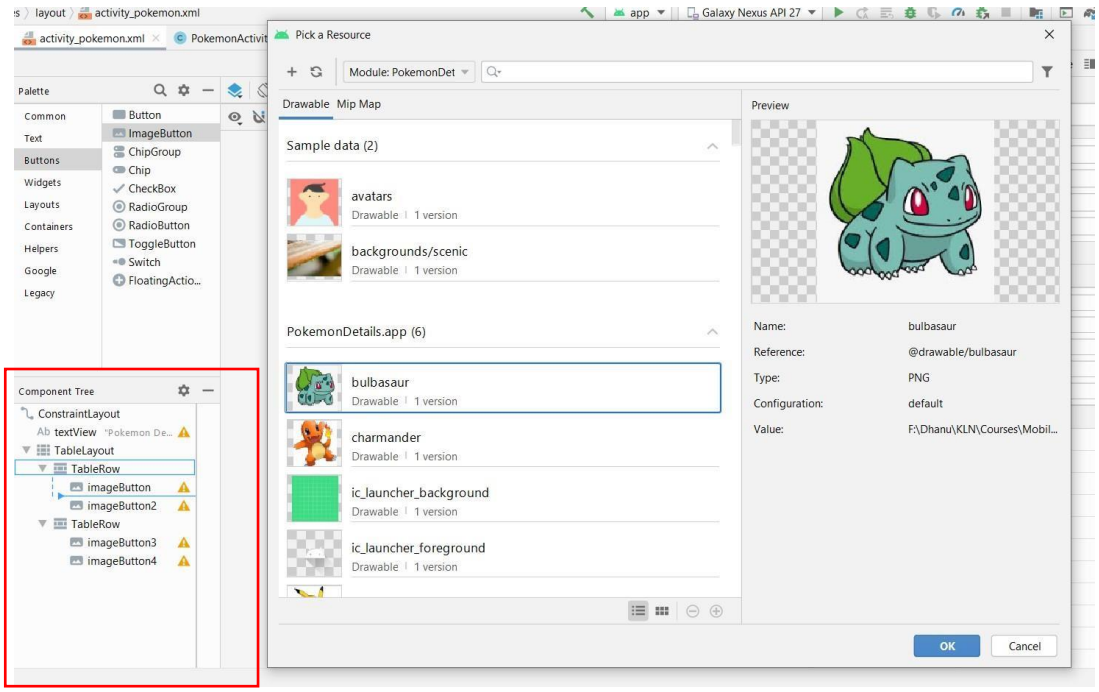
- Open android studio and create a new project with a “Empty Activity”.
- Give a name for the project, package name and save it in an appropriate folder. After the project has successfully build, open the “Android” project view.
- Download some cartoon images and their character details texts. Save the image and the details text files with matching name. Eg: “pikachu.png” and “pikachu.txt”. (I have used Pokemon characters so you can change the code names as suiting to your character names).
- Copy your images under to the “res/drawable” folder.
- Create a “raw” folder under the “res” directory and copy the character details text files to the “raw” folder.



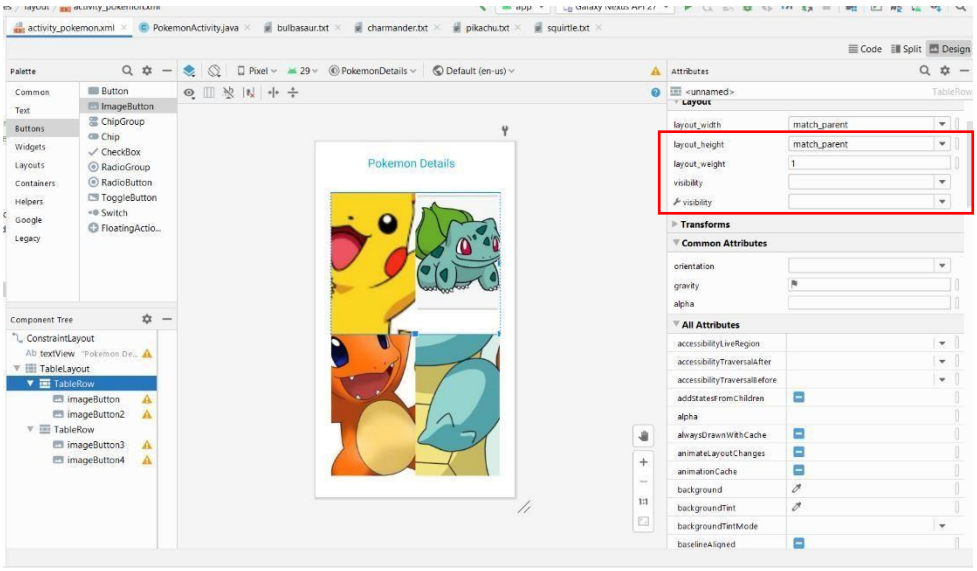
- Rename the “MainActivity.java” and “activity_main.xml” files to “PokemonActivity.java” and “activity_pokemon.xml” respectively.



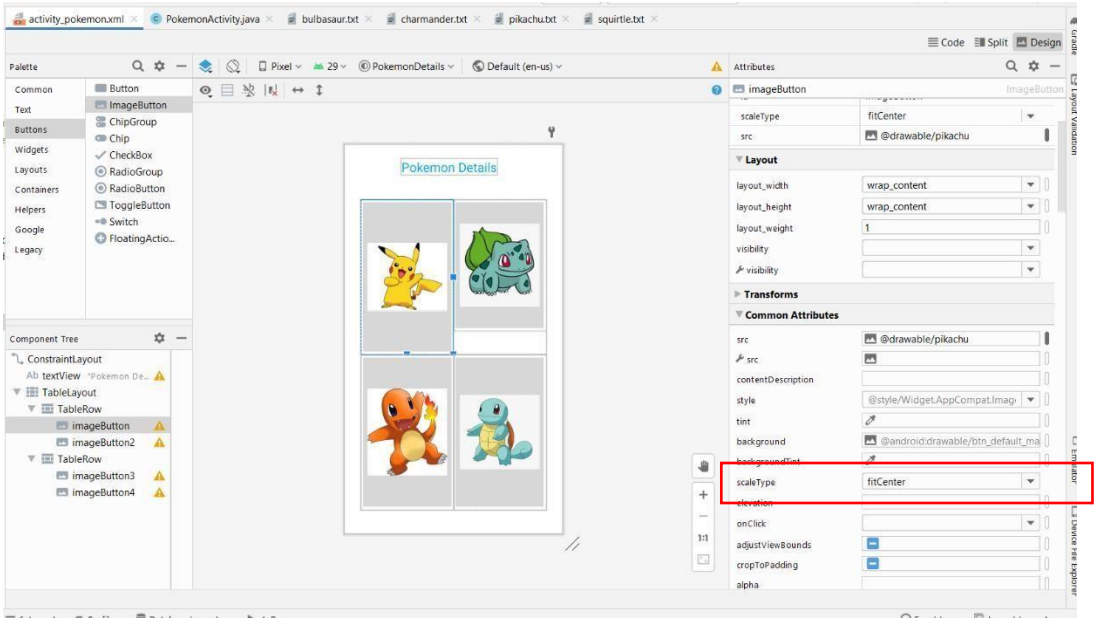
- Open the “activity_pokemon.xml” file and add a “TextView” widget and set its text as “Pokemon Details”. Position the widget on the top of the layout.
- Next add a “TableLayout” below the TextView widget and position with layout widget.
- Include two table rows to the table layout.
- Drag and drop an “ImageButton” widget to the Table layout. It will prompt a “Pick a resource” pop up. You can include an image you have copied to the project.
- To each row include two ImageButton widgets. To include the widget under the TableRow drag and drop the widget under the TableRow in the “Component Tree” panel.



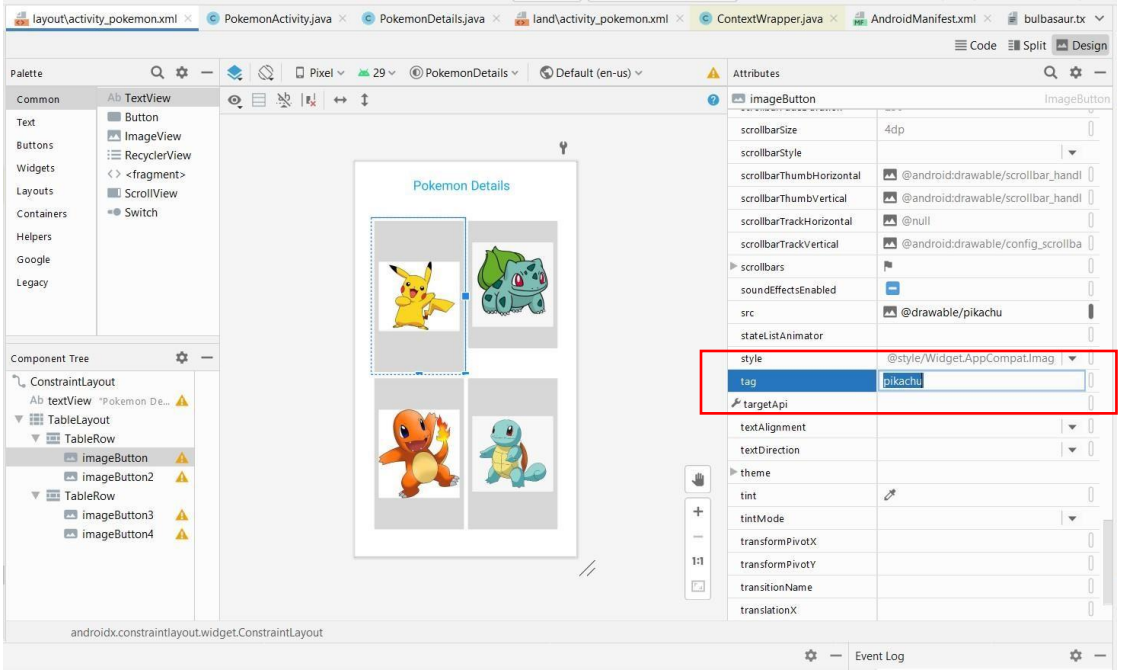
Set the “layout_weight” attribute of each row and image button to 1. Then the widgets will get positioned on the layout properly.



- To align the images properly set the “scaleType” attribute of each image to “fitCenter”.



Add the attribute tag to all the ImageButton widgets. The tag name should match the image and text file name you have given.

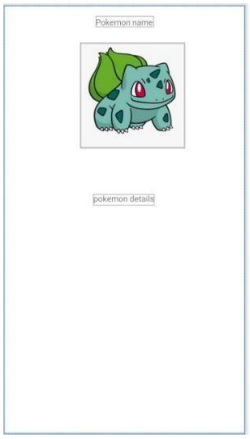


- Next open the “PokemonActivity.java” class and write click event method names “clickPokemon”. Attach this method to the “ImageButton” widgets under their onclick attribute.

- Create a new activity named “PokemonDetails.java”.

- Open the “activity_pokemon_details.xml” file.

- Include a TextView widget on top to display the pokemon name and give its ID as **pokemon_name**. Then below it include an ImageView and give it an already save image from your drawable folder. Give the id for the image view as **pokemon_img**. Below the image include another TextView to display the character details. Give the id for the TextView as **pokemon_details**.



Then open the “PokemonDetails.java” class and in the onCreate method write the following code logic. First you need to identify if a result from an intent got passed to the activity. For that you can add a condition as “intent != null”.

- Next after checking the intent we can get the content passed through the intent. Using “getStringExtra” or “getIntExtra” machines. After retrieving the extra detail using that we can retrieve the details and the images from the application resources. (You need to make sure the picture name, the details text and the name you want to display for the character are all the same).

```
class PokemonDetails : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_pokemon_details)

        if(intent != null){
            var pokemonName = intent.getStringExtra( name: "picName")

            var image: Int = resources.getIdentifier(pokemonName, defType: "drawable",packageName)
            var textID: Int = resources.getIdentifier(pokemonName, defType: "raw",packageName)
            var pokemonDetails: String = Scanner(resources.openRawResource(textID)).nextLine()

            var name = findViewById<TextView>(R.id.pokemon_name)
            var details = findViewById<TextView>(R.id.pokemon_details)
            var imageView = findViewById<ImageView>(R.id.pokemon_image)

            name.text = pokemonName
            details.text = pokemonDetails
            imageView.setImageResource(image)

        }
    }

}
```

Get image from drawable folder.

Get text file identifier from raw folder

Read the text file using the identifier.

- Now go back to the “PokemonActivity.java” class and write the logic inside the clickPokemon method.
- First get access to the ImageButton widget using findViewById method. Next get the tag attribute of the Image button. Pass this tag name through an intent to the PokemonDetails.java activity.

```
fun clickPokemon(view : View){
    var imageButton = findViewById<ImageButton>(view.id)
    var picName = imageButton.tag.toString()

    var intent = Intent( packageContext: this,PokemonDetails::class.java).apply {  this: Intent
        putExtra( name: "picName",picName)
    }
    startActivity(intent)
}
```