

SYMBIOSIS UNIVERSITY OF APPLIED SCIENCES (SUAS)

**India’s 1st Skill Development University**

**SKILL & PRACTICAL JOURNAL**

Enrollment Number: \_\_\_\_\_\_2019BTCS050\_\_\_\_\_\_\_ Year of Enrollment: \_\_\_\_\_\_\_\_\_2019-2023\_\_\_\_\_\_\_

Name of the Student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Navneet Mishra\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

School: \_\_\_\_\_\_\_\_\_\_Computer Science and Information Technology\_\_\_\_\_\_\_\_\_\_

Program: \_\_\_\_B-TECH\_\_\_\_

Specialization/ Branch: \_\_\_CSIT\_\_\_

Semester: \_\_\_IV\_\_\_\_ Section: \_\_\_\_\_\_B\_\_\_\_\_\_ Branch: \_\_\_\_\_B1\_\_\_\_\_\_

Paper Code: \_\_\_BTCS04CCB5\_\_\_\_ Name of Paper: \_\_Mobile Application Development\_\_

Faculty In-Charge: \_\_\_\_Mr. Deepak Yadav \_\_\_

**CERTIFICATE**

**THE SKILL AND PRACTICAL EXPERIMENT ENTERED IN THIS JOURNAL HAVE BEEN SATISFACTORILY PERFORMED BY**

**ENROLLMENT NO\_\_\_2019BTCS050\_\_ MR/MS\_\_NAVNEET MISHRA\_\_ STUDYING IN PROGRAM\_\_B-TECH\_\_ BRANCH\_\_CSIT\_\_\_ IN SCHOOL OF\_\_Computer Science and Information Technology\_\_\_ DURING SEMESTER\_\_\_IV\_\_ OF ACADEMIC YEAR\_\_\_ 2020-2021\_\_\_.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**( )**

**DATE: \_\_\_\_\_\_\_\_\_\_\_\_\_**

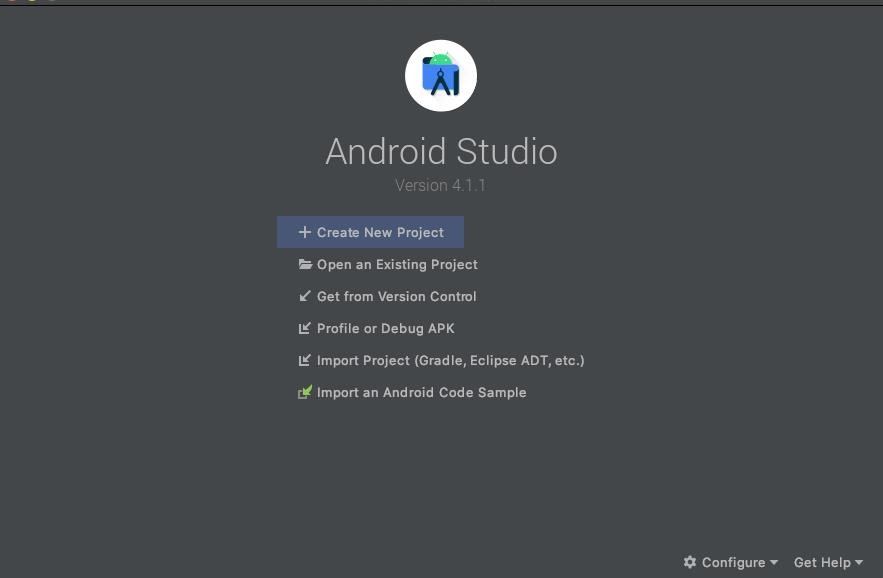
**INDEX**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Title of Practical Activity** | | **Date of Allocation** | **Date of Submission** | **Sign of Faculty** |
| 1) | Create Android Project. | | 10-05-2021 | 21-05-2021 |  |
| 2) | Identify and understand features of system libraries and files related to the installed  framework. | | 10-05-2021 | 21-05-2021 |  |
| 3) | Understand and link activities and intents, apply Styles and themes to activities. (Messenger App) | | 10-05-2021 | 21-05-2021 |  |
| 4) |  | Design an App | 10-05-2021 | 21-05-2021 |  |
| with different layout, having 8-  10 components. One layout(GridLayout App) is done during class, others has to be done by  you(student). | |
| 5) | Design an App for Career Advise, based on the input(Subject Area) it should suggest career choice(Courses). (CareerAdvisor  App) | | 10-05-2021 | 21-05-2021 |  |

**Practical 1: Create Android Project.**

To create your new Android project, follow these steps:

1. Install the latest version of Android Studio.
2. In the **Welcome to Android Studio** window, click **Create New Project**.



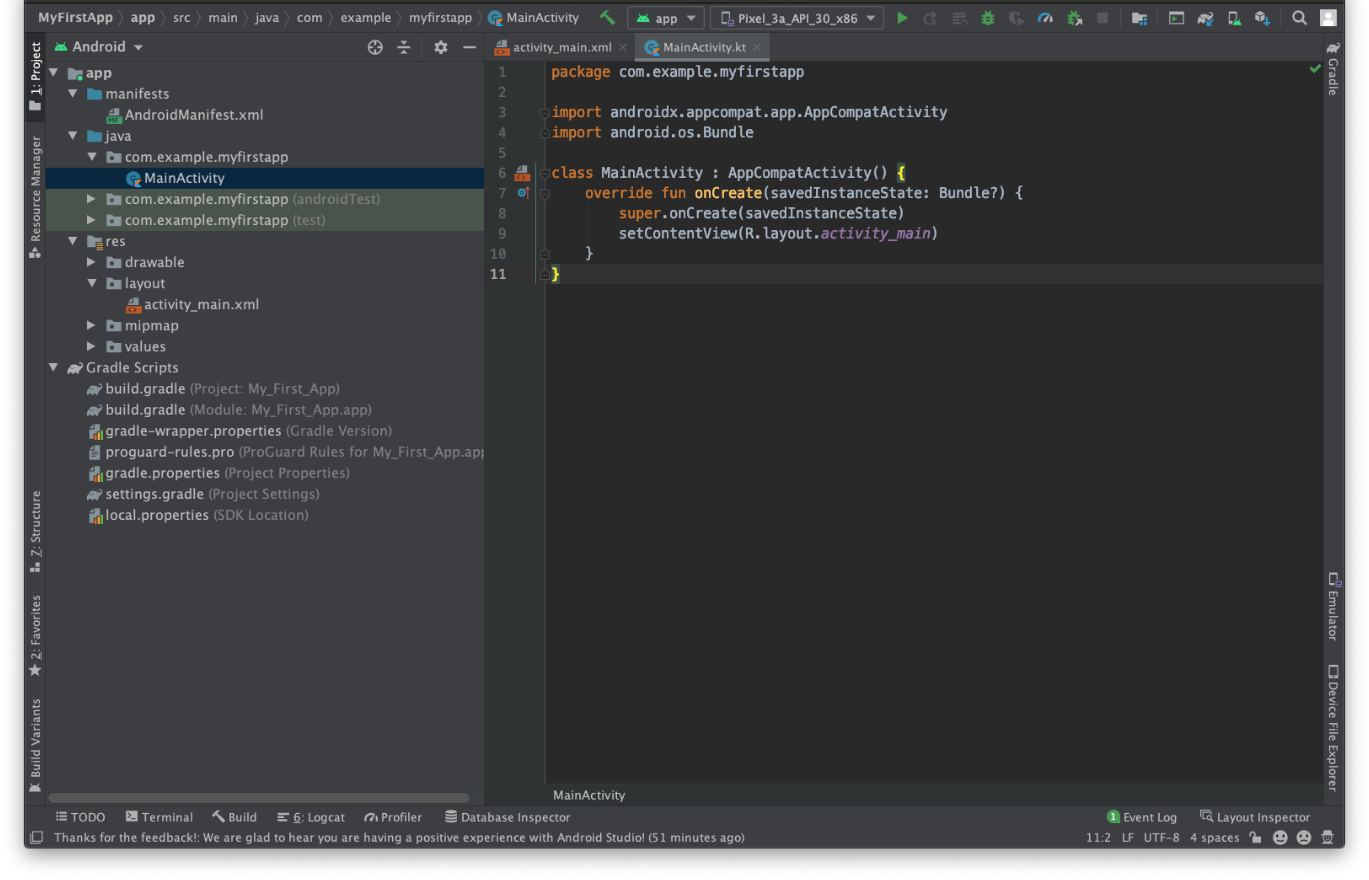
**Figure 1.** Android Studio welcome screen.

If you have a project already opened, select **File > New > New Project**.

In the **Select a Project Template** window, select **Empty Activity** and click **Next**.

1. In the **Configure your project** window, complete the following:
   * Enter "My First App" in the **Name** field.
   * Enter "com.example.myfirstapp" in the **Package name** field.
   * If you'd like to place the project in a different folder, change its **Save** location.
   * Select either **Java** or **Kotlin** from the **Language** drop-down menu.
   * Select the lowest version of Android you want your app to support in the **Minimum SDK** field.
   * If your app will require legacy library support, mark the **Use legacy android. Support libraries** checkbox.
   * Leave the other options as they are.
2. Click **Finish**.

After some processing time, the Android Studio main window appears.



**Figure 2.** Android Studio main window.

Now take a moment to review the most important files.

First, be sure the **Project** window is open (select **View > Tool Windows > Project**) and the Android view is selected from the drop-down list at the top of that window. You can then see the following files:

**app > java > com.example.myfirstapp > MainActivity**

This is the main activity. It's the entry point for your app. When you build and run your app, the system launches an instance of this Activity and loads its layout.

**app > res > layout > activity\_main.xml**

This XML file defines the layout for the activity's user interface (UI). It contains a TextView element with the text "Hello, World!"

**app > manifests > AndroidManifest.xml**

The manifest file describes the fundamental characteristics of the app and defines each of its components.

**Gradle Scripts > build.gradle**

There are two files with this name: one for the project, "Project: My\_First\_App," and one for the app module, "Module: My\_First\_App.app." Each module has its

own build.gradle file, but this project currently has just one module. Use each module's build.gradle file to control how the Gradle plugin builds your app.

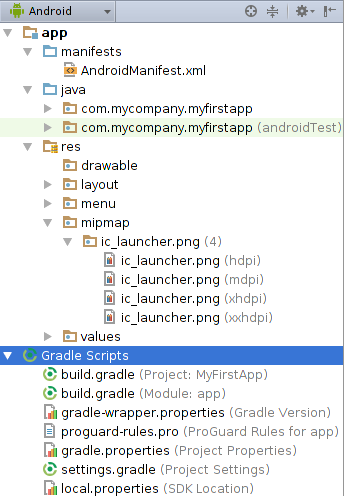
# Practical 2:Identify and understand features of system libraries and files related to the installed framework.

Library module

Provides a container for your reusable code, which you can use as a dependency in other app modules or import into other projects. Structurally, a library module is the same as an app module, but when built, it creates a code archive file instead of an APK, so it can't be installed on a device.

In the **Create New Module** window, Android Studio offers the following library modules:

* + Android Library: This type of library can contain all file types supported in an Android project, including source code, resources, and manifest files. The build result is an Android Archive (AAR) file that you can add as a dependency for your Android app modules.
  + Java Library: This type of library can contain only Java source files. The build result is an Java Archive (JAR) file that you can add as a dependency for your Android app modules or other Java projects.



By default, Android Studio displays your project files in the **Android** view. This view does not reflect the actual file hierarchy on disk, but is organized by modules and file types to simplify navigation between key source files of your project, hiding certain files or directories that are not commonly used. Some of the structural changes compared to the structure on disk include the following:

* Shows all the project's build-related configuration files in a top-level **Gradle Script** group.
* Shows all manifest files for each module in a module-level group (when you have different manifest files for different product flavours and build types).
* Shows all alternative resource files in a single group, instead of in separate folders per resource qualifier. For example, all density versions of your launcher icon are visible side-by-side.

Within each Android app module, files are shown in the following groups:

**manifests**

Contains the AndroidManifest.xml file.

**java**

Contains the Java source code files, separated by package names, including JUnit test code.

**res**

Contains all non-code resources, such as XML layouts, UI strings, and bitmap images, divided into corresponding sub-directories. For more information about all possible resource types, see Providing Resources.

**Practical 3: Understand and link activities and intents, apply styles and themes to activities. (Messenger App)**

The Activity class is a crucial component of an Android app, and the way activities are launched and put together is a fundamental part of the platform's application model. Unlike programming paradigms in which apps are launched with a main () method, the Android system initiates code in an Activity instance by invoking specific call-back methods that correspond to specific stages of its lifecycle.

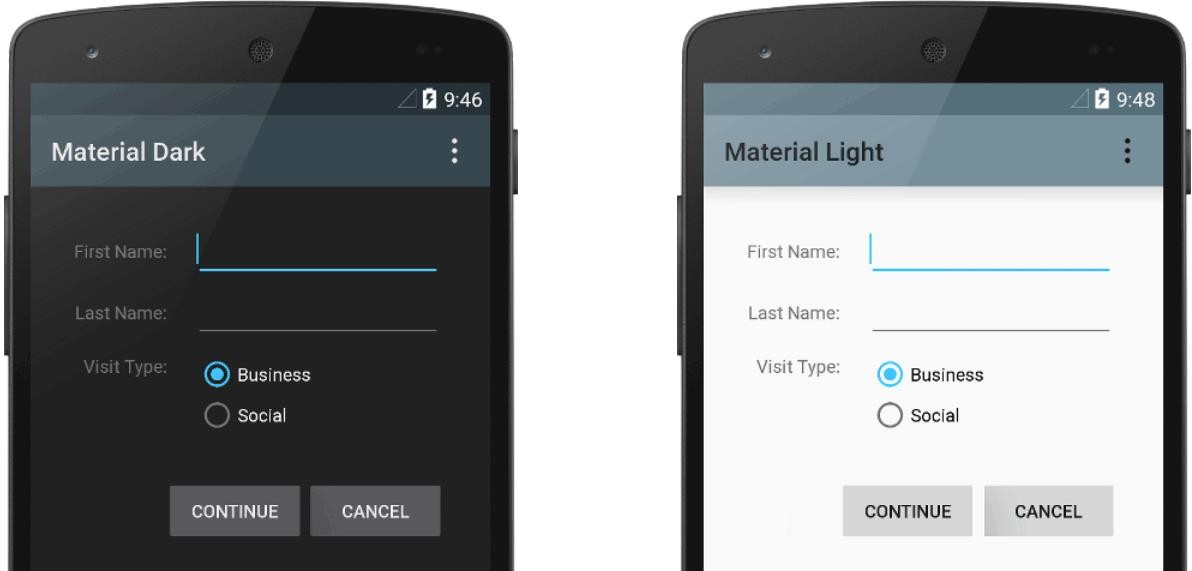
An Intent is a messaging object you can use to request an action from another app component. Although intents facilitate communication between components in several ways, there are three fundamental use cases:

Styles and themes on Android allow you to separate the details of your app design from the UI structure and behavior, similar to stylesheets in web design.

A *style* is a collection of attributes that specify the appearance for a single [View](https://developer.android.com/reference/android/view/View). A style can specify attributes such as font color, font size, background color, and much more.

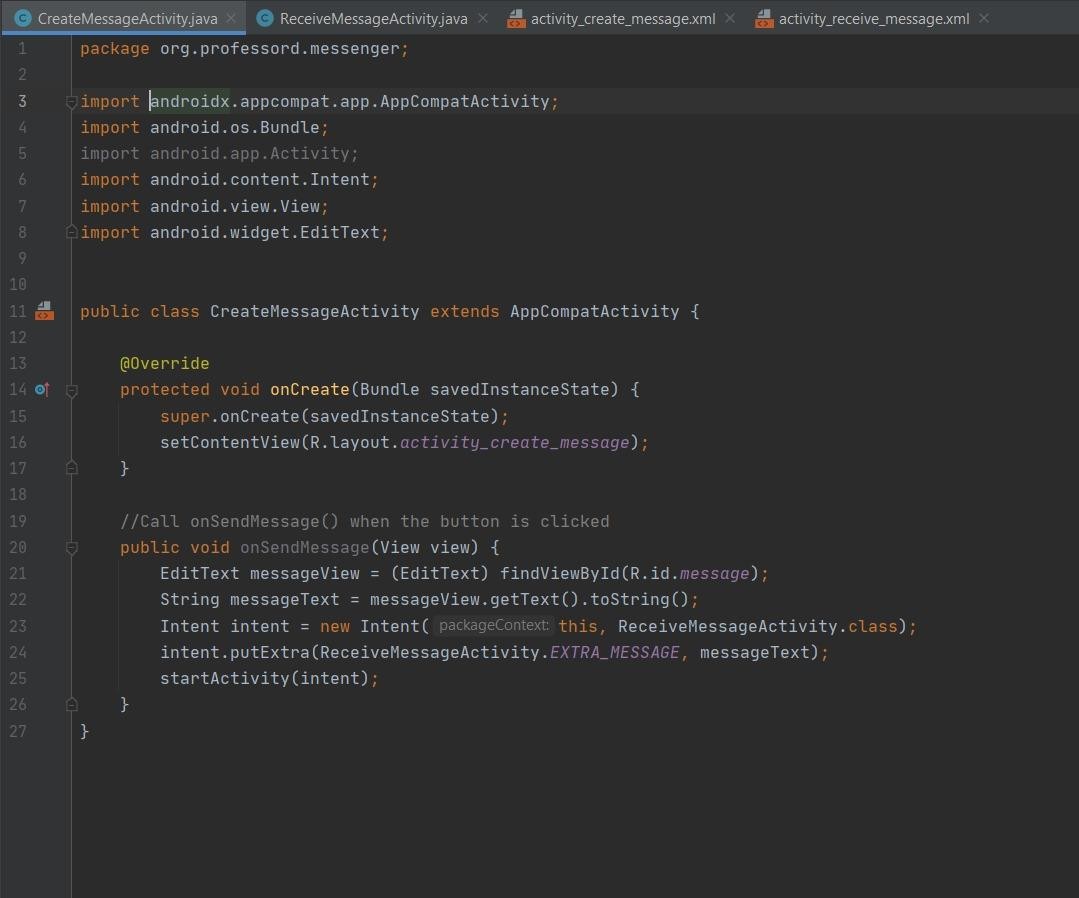
A *theme* is a collection of attributes that's applied to an entire app, activity, or view hierarchy—not just an individual view. When you apply a theme, every view in the app or activity applies each of the theme's attributes that it supports. Themes can also apply styles to non-view elements, such as the status bar and window background.

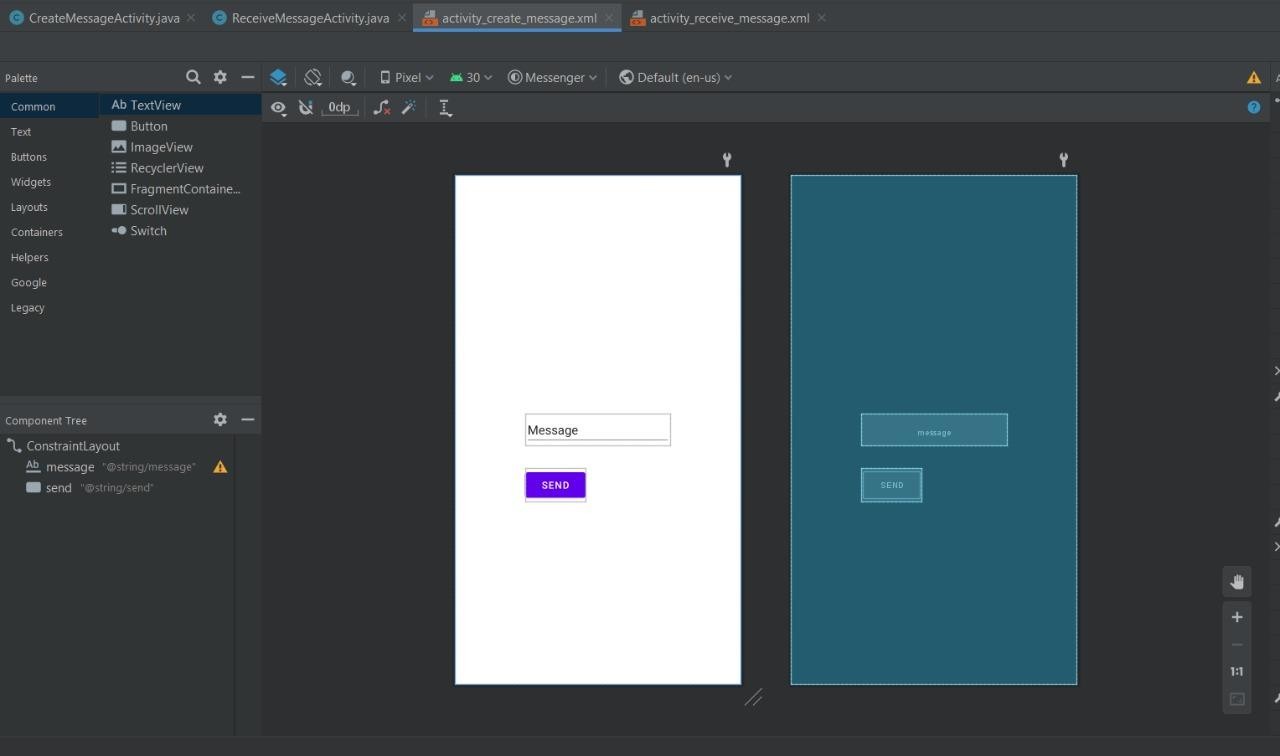
Styles and themes are declared in a style resource file in res/values/, usually named styles.xml.

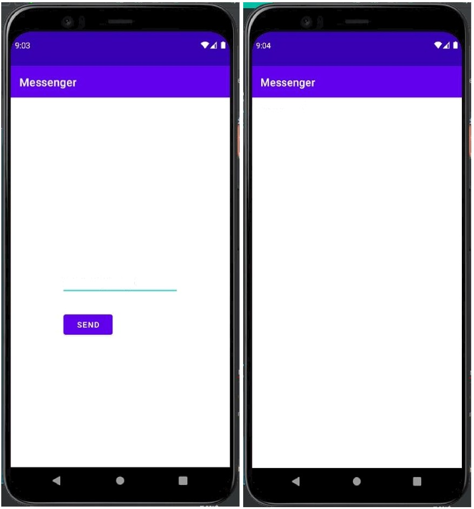


# (Messaging app)

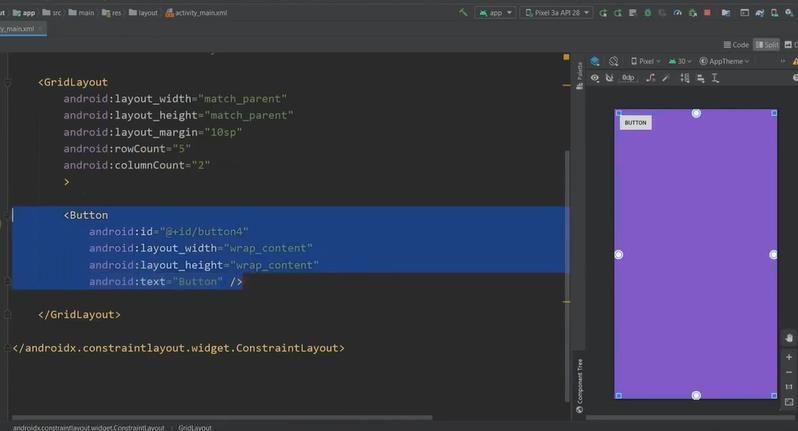
# 

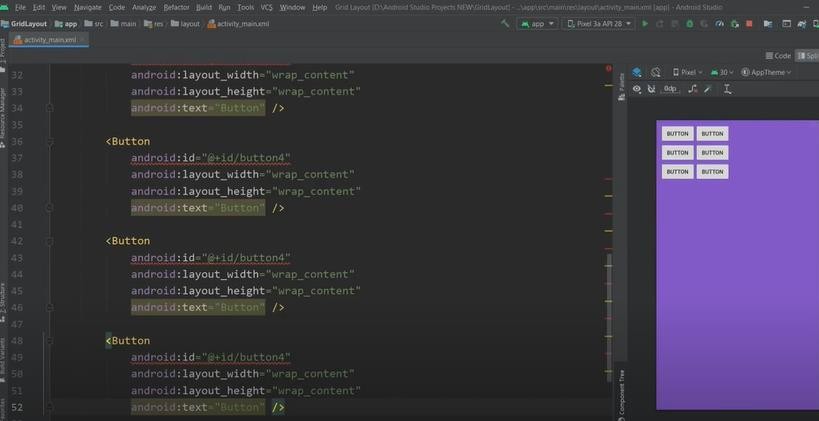


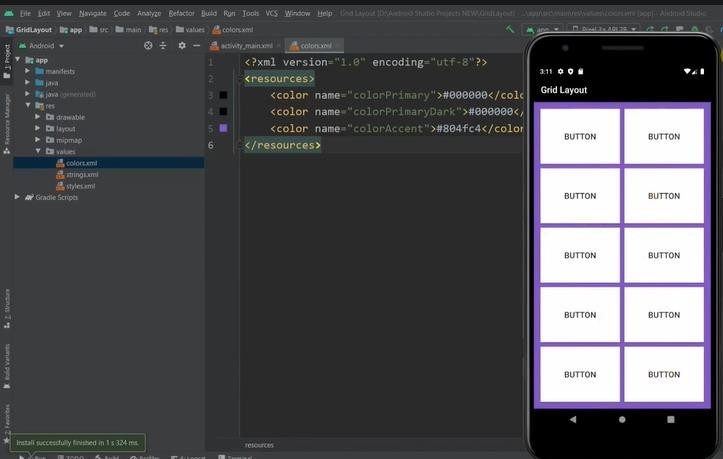


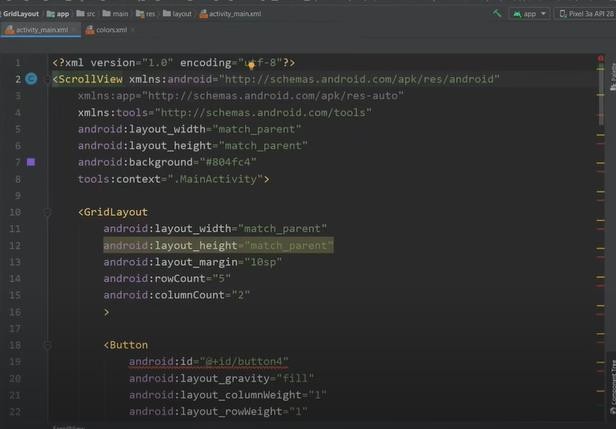


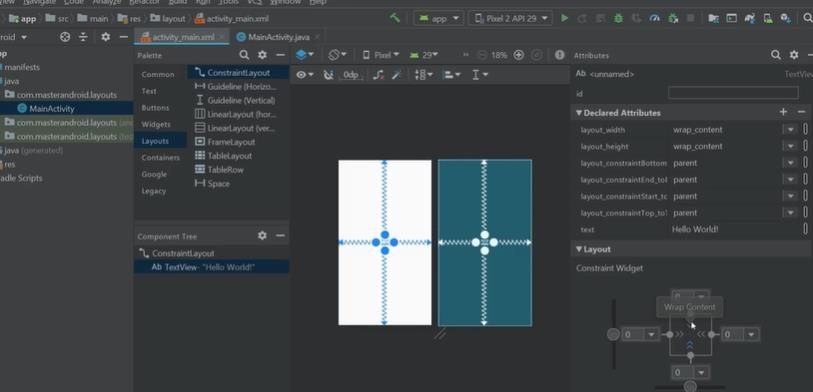
# Practical 4: Design an app with different layout having 8-10 components.

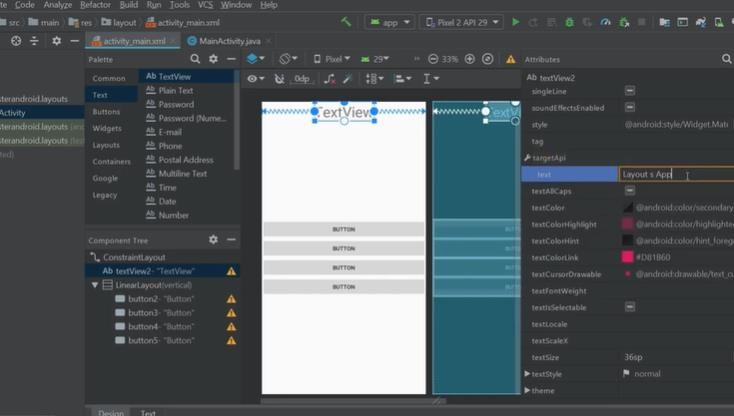


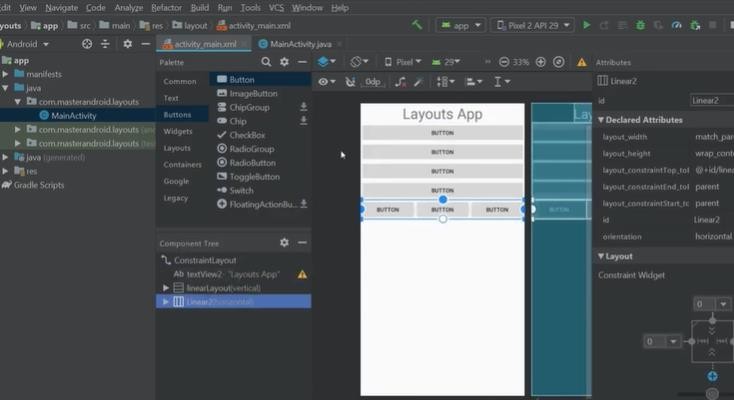












**Practical 5: Design an app for Career Advice, based on the input (subject area); it should suggest career choices.**

