INTERNSHIP REPORT

An internship report submitted in partial fulfilment of the requirements of IV B.Tech I Semester of

## BACHELOR OF TECHNOLOGY

**in**

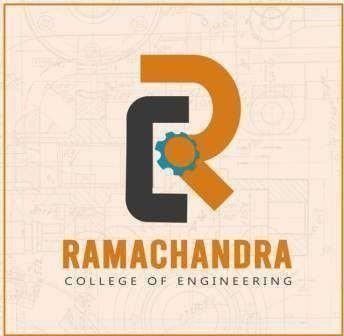
## ARTIFICIAL INTELLIGENCE & DATA SCIENCE

**Submitted By**

**Y. V. Shivani (20ME1A5464)**

**Under Supervision of**

**Mr. Sd. Arief Assistant Professor, AI&DS**



# RAMACHANDRA COLLEGE OF ENGINEERING(A)

**NH-16 Bypass Road, Vatluru(V), ELURU-534007, A.P. Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, KKD**

**Recognized by UGC 2(f) & 12(b)**

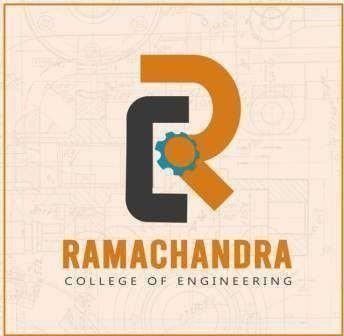
**Accredited by NAAC A+, NBA, ISO 9001: 2015 Certified**

**2023-2024**

# Department of Artificial Intelligence & Data Science



## DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE



**CERTIFICATE**

This is to certify that the “**Internship Report**” submitted by **Y. V. Shivani** Regd. No: **20ME1A5464** is work done by him/her and submitted during 2023-2024 Academic Year in partial fulfilment of the requirements of IV B.Tech I Semester of Bachelor of Technology in Artificial Intelligence and Data Science.

**Mr. Sd. Arief Dr. G. Chamundeswari**

**Assistant Professor, AI&DS Professor**

**Internship Supervisor HOD, AI&DS**

**External Examiner**

**ACKNOWLEDGEMENT**

I would like to take the opportunity to express our deep gratitude to all the people who have extended their cooperation in various ways during my internship. It is my pleasure and responsibility to acknowledge the help of all those individuals.

I would like to thank **Mr. Sd. Arief**, Assistant Professor for giving guidance and support to complete the internship.

I am very grateful to **Dr. G. Chamundeshwari**, Head of the Department, Department of Artificial Intelligence and Data Science for her guidance and encouragement in all respects in carrying throughout my internship.

I would like to express my sincere gratitude to **Dr. M. Srinivas Rao**, Principal, Ramachandra College of Engineering, Eluru for his valuable suggestions during preparation of draft in our document.

I express my heartful gratitude to the Management of Ramachandra College of Engineering, Eluru for their support and encouragement in completing my internship and providing me necessary facilities.

I sincerely thank all the faculty members and staff of the Department of AI&DS for their valuable advices, suggestions and constant encouragement which played a vital role in carrying out my internship.

Finally, I thank one and all who directly or indirectly helped me to complete my internship successfully.

**Y. V. Shivani**

**20ME1A5464**

# Declaration

We hereby declare that the **internship** on **“Android Application Development using Kotlin”** submitted by us to Jawaharlal Nehru Technological University Kakinada in partial fulfilment of the requirements of IV B. Tech I Semester of Bachelor of Technology in Artificial Intelligence & Data Science. This internship work carried out by us under the supervision of **Mr. Sd. Arief,** Assistant Professor in AI&DS.

**Y. V. Shivani 20ME1A5464**

**Abstract**

This Android Application Development with Kotlin Internship Program will provide the skills you need to

become an Android Application Developer with ample hands-on experience and dedicated Mentor support.

As world’s technology is rapidly growing, we have fast connection and network to instantly connect to other

person. Day to day use in mobile, tablets and laptop is increasing, most of the people already have this

facilities. In this fast and information-oriented world we need to stay updated with every incidents and news

too. This News app is android mobile application where user have access to latest news from 120+

newspapers from 50+ countries. The main focus of this application is to connect news articles from all around

the world and deliver it to user as fast as possible in best visualize way.

Kotlin is a modern statically typed programming language used by over 60% of professional Android

developers that helps boost productivity, developer satisfaction, and code safety.

Kotlin is an open source project available at no charge under the Apache 2.0 license. The code for the project

is developed openly on GitHub primarily by the team employed at JetBrains, with contributions from Google

and others. Our choice of Kotlin reaffirms our commitment to an open developer ecosystem as we evolve and

grow the Android platform, and we are excited to see the language evolve.

BM SkillsBuild 6 Weeks Internship in Emerging Technologies in Artificial Intelligence

The IBM SkillsBuild 6 Weeks Internship for Emerging Technologies in Artificial Intelligence

offers a transformative learning experience for individuals aspiring to excel in the field of AI.

Spanning six weeks, the program immerses participants in comprehensive learning tracks that

encompass key facets of AI, including machine learning, natural language processing, and

computer vision. Through hands-on projects and expert mentorship, participants gain practical

insights into real-world AI applications. The internship fosters collaborative learning, enabling

interaction with peers and industry professionals. Upon successful completion, participants

receive a certification that validates their acquired skills and prepares them for the dynamic

landscape of AI. The program is tailored for individuals at all skill levels, making it an inclusive

opportunity for exploration and advancement in the realm of emerging AI technologies

The IBM SkillsBuild 6 Weeks Internship for Emerging Technologies in Artificial Intelligence

offers a transformative learning experience for individuals aspiring to excel in the field of AI.

Spanning six weeks, the program immerses participants in comprehensive learning tracks that

encompass key facets of AI, including machine learning, natural language processing, and

computer vision. Through hands-on projects and expert mentorship, participants gain practical

insights into real-world AI applications. The internship fosters collaborative learning, enabling

interaction with peers and industry professionals. Upon successful completion, participants

receive a certification that validates their acquired skills and prepares them for the dynamic

landscape of AI. The program is tailored for individuals at all skill levels, making it an inclusive

opportunity for exploration and advancement in the realm of emerging AI technologies.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Contents** | **Page No** |
| 1 | Internship Certificate | 7 |
| 2 | Introduction to Company/Institution | 8 |
| 3 | Learning Objectives/Internship Objectives | 9 |
| 4 | Weekly overview of internship activities | 10-11 |
| 5 | Introduction to Internship Topic | 12-14 |
| 6 | Modules | 15 |
| 7 | Description of Internship | 16-17 |
| 8 | Build your first Android app in Compose | 18-19 |
| 9 | Build an interactive Android app | 20-36 |
| 10 | Build Lists and Add Theme | 36 |
| 11 | Navigation and App Architecture | 37 |
| 12 | Connect to the Internet | 37 |
| 13 | Data Persistence | 37 |
| 14 | Work Manager | 38 |
| 15 | Compose with Views | 38 |
| 16 | Description of Project | 39-40 |
| 17 | Implementation of Project | 41-42 |
| 18 | Screenshots | 43-46 |
| 19 | Reflection on the Internship | 47 |
| 20 | Conclusion | 48 |

1. **Internship Certificate:**



1. **Introduction to Company/Institution**

At **SmartBridge**, our cutting-edge edtech platform, "SmartInternz," serves as a catalyst for fostering collaboration between academia and industry. By providing project-based, collaborative learning solutions intricately woven into the curriculum, it empowers students to cultivate the essential technical and professional skills required to become job-ready candidates. The platform's immersive learning journey equips students with the necessary expertise to excel in their chosen careers.

SmartBridge is dedicated to a momentous talent mission: to provide "1 Million Virtual Internships" across a wide range of in-demand technologies. Our goal is twofold: to assist companies in finding job-ready talent and to play a pivotal role in building a thriving gig economy in India.

## Our Objective:

Our main objective is to bridge the existing gaps between prevailing industry standards and what the academics offer to the graduates while passing out of university. SmartBridge offers suitable skill deployment and training to the young talent before on boarding their first job.

Our skill development programs are designed considering the present in demand skills in the industry. We thereby work along the lines to offer best programs that helps the students to gain practical knowledge and hands on training to learn the skills of the future.

### [**Main Objectives Of SmartBridge:**](https://www.thesmartbridge.com/Aboutus)

* Internship for Every Student
* Promote Industry Approved Professional Electives
* Become a Talent Factory of India by 2026

## Our Core Values:

### [Student-Centric Approach](https://www.thesmartbridge.com/Aboutus)

Our students are at the core of everything we do. We prioritize their learning needs, aspirations, and career growth, providing personalized support and guidance.

### [Equal Opportunities](https://www.thesmartbridge.com/Aboutus)

We are committed to ensuring equal opportunities for all students, regardless of their geographical location. We strive to bridge the gap between students studying in cities and remote areas, empowering them with the same level of access to quality education and opportunities.

### [Outcome-Driven Partnerships](https://www.thesmartbridge.com/Aboutus)

We believe in forging partnerships that are focused on tangible outcomes and mutual success. Our collaborations are geared towards achieving concrete results and positive impact.

### [Innovation](https://www.thesmartbridge.com/Aboutus)

Embracing innovation is fundamental to our ethos. We constantly seek new and effective ways to enhance learning experiences, staying at the forefront of emerging technologies and methodologies.

### [Social Impact](https://www.thesmartbridge.com/Aboutus)

We are driven by a sense of responsibility to make a positive impact on society. Our efforts go beyond individual success stories, seeking to uplift communities and contribute to a better world.

1. **Learning Objectives / Internship Objectives**

* Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.
* An objective for this position should emphasize the skills you already possess in the area and your interest in learning more.
* Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
* Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
* Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

1. **WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Day** | **Name of Topic / Module Completed** |
| I | 14/06/2023 | Monday | Online Session – Explanation of Internship Program |
| 15/06/2023 | Tuesday | Online Session – Account Creation |
| 16/06/2023 | Wednesday | Online Session - Account Creation |
| 17/06/2023 | Thursday | Self-pace Learning – Module1 |
| 18/06/2023 | Friday | Self-pace Learning – Module1 |
| 19/06/2023 | Saturday | Self-pace Learning – Module1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Day** | **Name of Topic / Module Completed** |
| II | 21/06/2023 | Monday | Self-pace Learning – Module2 |
| 22/06/2023 | Tuesday | Self-pace Learning – Module2 |
| 23/06/2023 | Wednesday | Self-pace Learning – Module2 |
| 24/06/2023 | Thursday | Self-pace Learning – Module3 |
| 25/06/2023 | Friday | Self-pace Learning – Module3 |
| 26/06/2023 | Saturday | Self-pace Learning – Module3 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Day** | **Name of Topic / Module Completed** |
| III | 28/06/2023 | Monday | Self-pace Learning – Module4 |
| 29/06/2023 | Tuesday | Self-pace Learning – Module4 |
| 30/06/2023 | Wednesday | Self-pace Learning – Module4 |
| 01/07/2023 | Thursday | Self-pace Learning – Module5 |
| 02/07/2023 | Friday | Self-pace Learning – Module5 |
| 03/07/2023 | Saturday | Self-pace Learning – Module5 |
| **Week** | **Date** | **Day** | **Name of Topic / Module Completed** |
| IV | 04/06/2023 | Monday | Self-pace Learning – Module6 |
| 05/06/2023 | Tuesday | Self-pace Learning – Module6 |
| 06/06/2023 | Wednesday | Self-pace Learning – Module6 |
| 07/07/2023 | Thursday | Self-pace Learning – Module7 |
| 08/07/2023 | Friday | Self-pace Learning – Module7 |
| 09/07/2023 | Saturday | Self-pace Learning – Module7 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Day** | **Name of Topic / Module Completed** |
| V | 10/07/2023 | Monday | Self-pace Learning – Module8 |
| 11/07/2023 | Tuesday | Self-pace Learning – Module8 |
| 12/07/2023 | Wednesday | Self-pace Learning – Module8 |
| 13/07/2023 | Thursday | Completion of Hands-on Project |
| 14/07/2023 | Friday | Online Session – Doubts + Q/A Session |
| 15/07/2023 | Saturday | Completion of Hands-on Project |

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Day** | **Name of Topic / Module Completed** |
| VII | 22/07/2023 | Monday | Project Validation |
| 23/07/2023 | Tuesday | Project Validation |
| 24/07/2023 | Wednesday | Project Validation |
| 25/07/2023 | Thursday | Project Validation |
| 26/07/2023 | Friday | Project Validation |
| 27/07/2023 | Saturday | Issuing Certificate of Completion Letter |

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Day** | **Name of Topic / Module Completed** |
| VIII | 28/07/2023 | Monday | Issuing Certificate of Completion Letter |
| 29/07/2023 | Tuesday | Issuing Certificate of Completion Letter |
| 30/07/2023 | Wednesday | Issuing Certificate of Completion Letter |
| 01/08/2023 | Thursday | Issuing Certificate of Internship |
| 02/08/2023 | Friday | Issuing Certificate of Internship |
| 03/08/2023 | Saturday | Issuing Certificate of Internship |
| 04/08/2023 | Sunday | Issuing Certificate of Internship |

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Date** | **Day** | **Name of Topic / Module Completed** |
| VI | 16/07/2023 | Monday | Completion of Hands-on Project |
| 17/07/2023 | Tuesday | Online Session – Submission Details + Q/A Session |
| 18/07/2023 | Wednesday | Completion of Hands-on Project + PPT |
| 19/07/2023 | Thursday | Completion of Hands-on Project + PPT |
| 20/07/2023 | Friday | Completion of Hands-on Project + PPT |
| 21/07/2023 | Saturday | Completion of Hands-on Project + PPT |

1. **Introduction to Internship Topic**

## Topic: Android Application Development using Kotlin

Android operating system is the largest installed base among various mobile platforms across the globe. Hundreds of millions of mobile devices are powered by Android in more than 190 countries of the world. It conquered around 71% of the global market share by the end of 2021, and this trend is growing bigger every other day. The company named Open Handset Alliance developed Android for the first time that is based on the modified version of the Linux kernel and other open-source software. Google sponsored the project at initial stages and in the year 2005, it acquired the whole company. In September 2008, the first Android-powered device was launched in the market. Android dominates the mobile OS industry because of the long list of features it provides. It’s user-friendly, has huge community support, provides a greater extent of customization, and a large number of companies build Android-compatible smartphones. As a result, the market observes a sharp increase in the demand for developing Android mobile applications, and with that companies need smart developers with the right skill set. At first, the purpose of Android was thought of as a mobile operating system. However, with the advancement of code libraries and its popularity among developers of the divergent domain, Android becomes an absolute set of software for all devices like tablets, wearables, set-top boxes, smart TVs, notebooks, etc.

**Features of Android:**

Android is a powerful open-source operating system that open-source provides immense features

and some of these are listed below.

Android is a powerful open-source operating system that open-source provides immense features and some of these are listed below.

* Android Open Source Project so we can customize the OS based on our requirements.
* Android supports different types of connectivity for GSM, CDMA, Wi-Fi, Bluetooth, etc. for telephonic conversation or data transfer.
* Using Wi-Fi technology we can pair with other devices while playing games or using other applications.
* It contains multiple APIs to support location-tracking services such as GPS.
* We can manage all data storage-related activities by using the file manager.
* It contains a wide range of media supports like AVI, MKV, FLV, MPEG4, etc. to play or record a variety of audio/video.
* It also supports different image formats like JPEG, PNG, GIF, BMP, MP3, etc.
* It supports multimedia hardware control to perform playback or recording using a camera and microphone.
* Android has an integrated open-source WebKit layout-based web browser to support User Interfaces like HTML5, and CSS3.
* Android supports multi-tasking means we can run multiple applications at a time and can switch between them.
* It provides support for virtual reality or 2D/3D Graphics.

**Programming Languages used in Developing Android Applications:**

1. Java
2. Kotlin

Developing the Android Application using Kotlin is preferred by Google, as Kotlin is made an official language for Android Development, which is developed and maintained by JetBrains.

Previously before Java is considered the official language for Android Development. Kotlin ismade official for Android Development in Google I/O 2017.

**Advantages of Android Development:**

* The Android is an open-source Operating system and hence possesses a vast community for support.
* The design of the Android Application has guidelines from Google, which becomes easier for developers to produce more intuitive user applications.
* Fragmentation gives more power to Android Applications. This means the application can run two activities on a single screen.
* Releasing the Android application in the Google play store is easier when it is compared to other platforms.

**Disadvantages of Android Development:**

* Fragmentation provides a very intuitive approach to user experience but it has some drawbacks, where the development team needs time to adjust to the various screen sizes of mobile smartphones that are now available in the market and invoke the particular features in the application.
* The Android devices might vary broadly. So the testing of the application becomes more difficult.
* As the development and testing consume more time, the cost of the application may increase, depending on the application’s complexity and features.

**Kotlin Android Basics:**

Kotlin is a statically typed, general-purpose programming language developed by JetBrains that has built world-class IDEs like IntelliJ IDEA, PhpStorm, Appcode, etc. It was first introduced by JetBrains in 2011 and a new language for the JVM. Kotlin is object-oriented language, and a “better language” than Java, but still be fully interoperable with Java code. Kotlin is sponsored byGoogle, announced as one of the official languages for Android Development in 2017.

countries of the world. It conquered around 71% of the global market share by the end of 2021,

and this trend is growing bigger every other day. The company named Open Handset

Alliance developed Android for the first time that is based on the modified version of the Linux

kernel and other open-source software. Google sponsored the project at initial stages and in the

year 2005, it acquired the whole company. In September 2008, the first Android-powered device

was launched in the market. Android dominates the mobile OS industry because of the long list of

features it provides. It’s user-friendly, has huge community support, provides a greater extent of

customization, and a large number of companies build Android-compatible smartphones. As a

result, the market observes a sharp increase in the demand for developing Android mobile

applications, and with that companies need smart developers with the right skill set. At first, the

purpose of Android was thought of as a mobile operating system. However, with the advancement

of code libraries and its popularity among developers of the divergent domain, Android becomes

an absolute set of software for all devices like tablets, wearables, set-top boxes, smart TVs,

notebooks, etc.

Features of Android

Android is a powerful open-source operating system that open-source provides immense features

and some of these are listed below.

1. **Modules**

1.Build your first Android app in Compose

2.Build an interactive Android app

3.Build Lists and Add Theme

4.Navigation and App Architecture

5.Connect to the Internet

6.Data Persistence

7.Work Manager

8.Compose with Views

1. **Description of Internship:**

SmartBridge and Google have teamed up to create an outcome-driven skilling effort that will train 2000+

educators and 5000 students on android application development in Kotlin programming. This program has

been recognised by AICTE for delivery as a virtual internship program to all higher education students in

India.

Program will be executed in two phases, phase-1 is to teach the educators on android skills and application

lifecycle management and qualify them as "Mentor on Campus" to drive the phase-2 of the program in

campus called virtual internship program. The virtual internship program is 100+ Hrs. experiential learning

program containing hands-on bootcamps, courses, learning resources and project work.

Successful learners will be receiving the virtual internship completion certificate. Also, they will get an

opportunity to join the Google developer community in India.

##### **ABOUT PROGRAM:**

As demand for skilled Android developers increases in the job marketplace, there is

an even greater need for educators to train the next generation of Android developers. That is why we created

these resources to help support and empower educators.

##### **LEARNING OUTCOME:**

* 100 Hrs. of experiential learning includes online bootcamp + Self-learning + Guided project.
* Hands-on practice with Kotlin programming language
* Develop an Android application
* Complete google course on Android Basics in Kotlin
* Receive verifiable virtual internship certificate
* Helps to attempt the professional certification

##### **TRAINING CURRICULUM**

* Step-by-Step procedure to develop mobile application
* Confirming the idea & market research - using Mural
* Wireframe design - using Figma
* Kotlin Basics
* Layouts
* Navigation
* Connect to the internet
* Data Persistence
* Work manager

**Projects:** Develop Android Applications powered by AI / ML API’s (Vision API’s, Natural Language API’s)

##### **PROGRAM SCHEDULE:**

Program Launch: 6 June 2022

Registration Ends on: 30 July 2022

Virtual Internship Starts on: 1 August 2022

Virtual Internship Ends on: 30 September 2022

Certificate Distribution by: 15 October 2022

1. **Build your first Android app in Compose**

**Introduction:**

Android is an open-source operating system that runs on the Linux kernel. With the advent

of new mobile technologies, the mobile application industry is advancing rapidly. Consisting of several

operating systems like Symbian OS, iOS, blackberry, etc., Android OS is recognized as the most widely used,

popular and user-friendly mobile platform. This open-source Linux kernel based operating system offers high

flexibility due to its customization properties making it a dominant mobile operating system.

Android applications are developed using the java language. Google has its own Software

Development Kit (SDK) which enables these java codes to control devices like mobile phones, tablets, etc.

Android mobile application development provides a flexible platform for developers where they can use both

java Integrated Development Environment (IDEs) and android java libraries

Google android SDK delivers a special software stack that provides developers an easy platform to develop

android applications. Moreover, developers can make use of existing java IDEs which provides flexibility to

the developers. Java libraries are predominant in the process of third-party application development. Cross-

platform approaches make sure that developers do not have to develop platform-dependent applications. With

the help of these approaches, an application can be deployed to several platforms without the need for changes

in coding. However, android is more prone to security vulnerabilities which the majority of the users do not

take into account. Any android developer can upload their application on the android market which can cause

a security threat to any android device. These applications do not have to go through rigorous security checks.

Android is an open source and Linux-based Operating System for mobile devices such as smartphones and

tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

Android offers a unified approach to application development for mobile devices which means developers need

only develop for Android, and their applications should be able to run on different devices powered by Android.

The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008. On June 27, 2012, at the Google

I/O conference, Google announced the next Android version, 4.1Jelly Bean. Jelly Bean is an incremental

update, with the primary aim of improving the user interface, both in terms of functionality and performance.

The source code for Android is available under free and open source software licenses. Google publishes most

of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General

Public License version 2.

[**Android Basics with Compose**](https://developer.android.com/courses/android-basics-compose/unit-1)

Android Basics with Compose is a self-paced, online course on how to build

Android apps using the latest best practices. It covers the basics of building apps with Jetpack Compose, the

recommended toolkit for building user interfaces on Android.

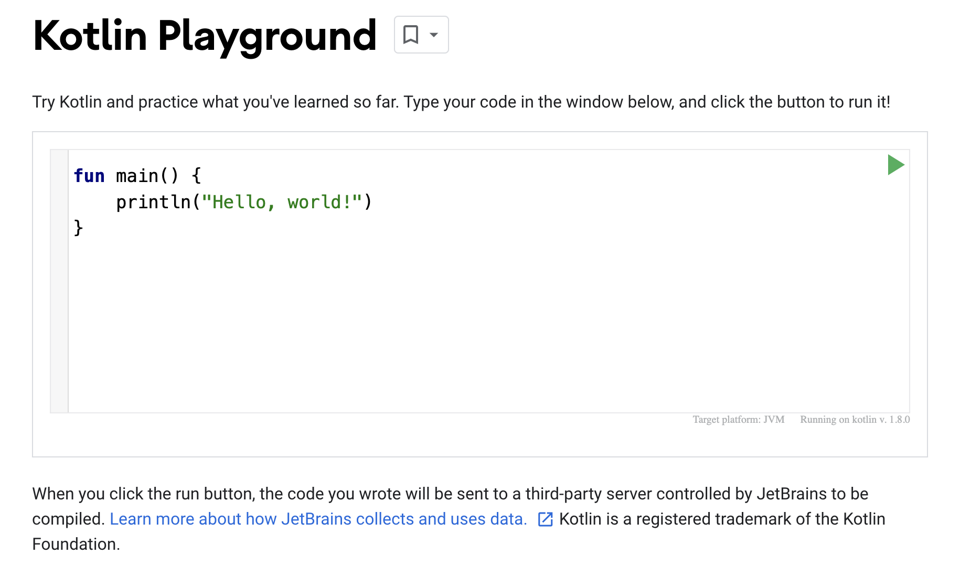
1. [**Build Your first Android App**](https://developer.android.com/courses/android-basics-compose/unit-1)

* Write simple Kotlin programs that display text output.
* Download and install Android Studio.
* Build an Android app with a simple user interface that displays text and images.
* Run the app on a device or emulator.

## [Open Kotlin Playground](https://developer.android.com/codelabs/basic-android-kotlin-compose-first-program?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-basics-compose-unit-1-pathway-1%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-compose-first-program#2)

In a web browser on your computer, open the [Kotlin Playground](https://developer.android.com/training/kotlinplayground).

You should see a web page similar to this image:



There's already some default code populated in the code editor. These three lines of code make up a simple program:

fun main() {  
    println("Hello, world!")  
}

## [Run your first program](https://developer.android.com/codelabs/basic-android-kotlin-compose-first-program?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-basics-compose-unit-1-pathway-1%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-compose-first-program#3)

Click Run button to run your program.

When you click the Run button, a lot happens. Code in the Kotlin programming language is meant to be understood by humans, so that people can more easily read, write, and collaborate on Kotlin programs. However, your computer doesn't immediately understand this language.

You need something called the Kotlin compiler, which takes the Kotlin code you wrote, looks at it line by line, and translates it into something that the computer can understand. This process is called compiling your code.

If your code compiles successfully, your program will run (or execute). When the computer executes your program, it performs each of your instructions. If you've ever followed a cooking recipe, performing each step in the recipe is considered executing each instruction.



At the bottom of the code editor, you should see a pane that shows the output, or result, of your program:

Hello, world!

Cool! The purpose of this program is to print or display a message that says Hello, world!.

## Create a project using the template

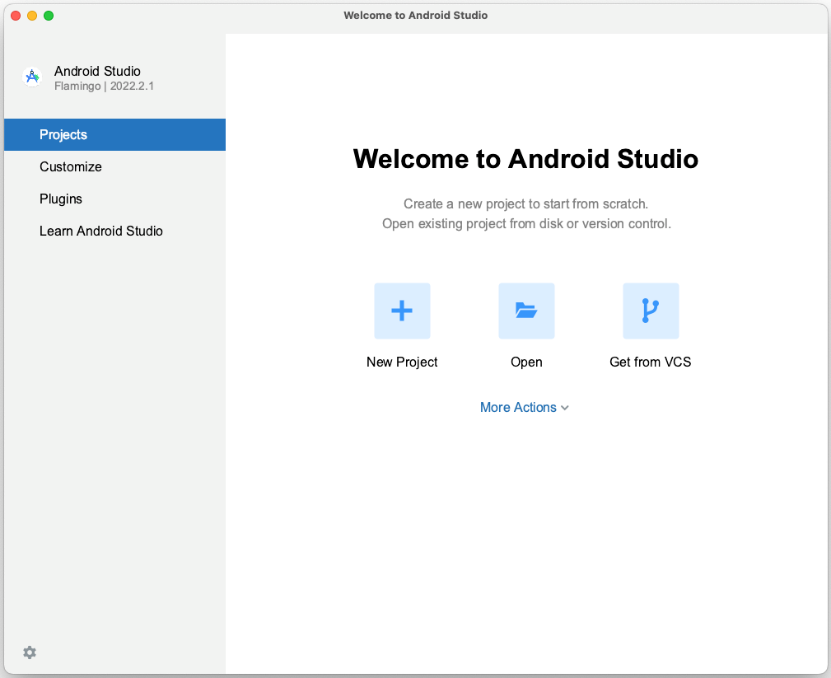
In this codelab, you create an Android app with the **Empty Activity** project template provided by Android Studio.

To create a project in Android Studio:

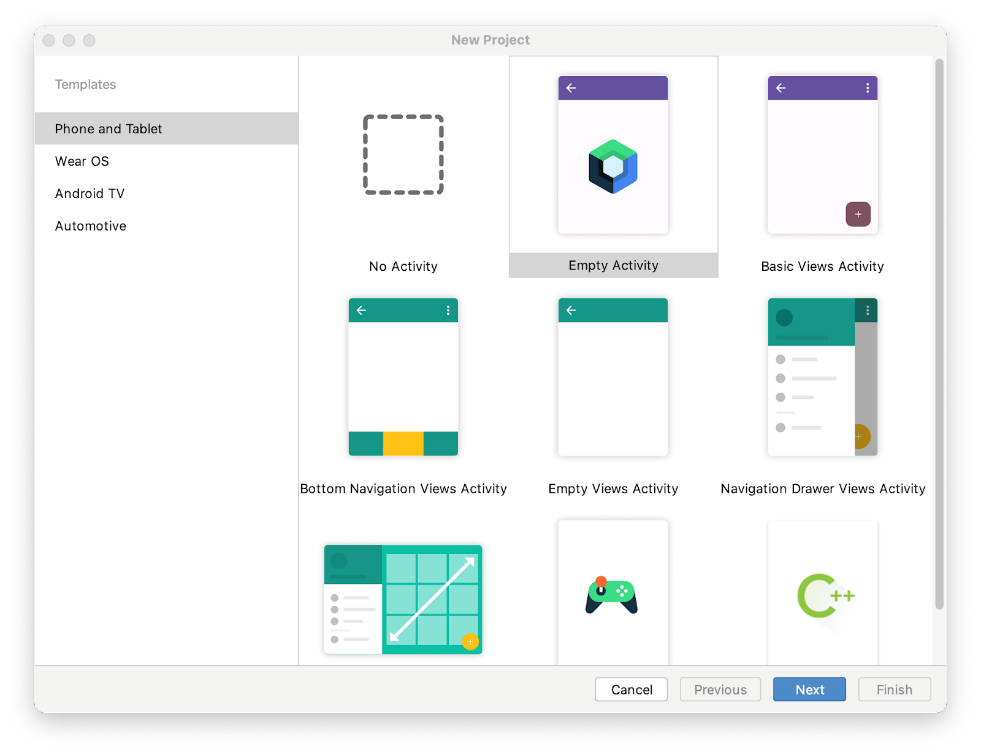
1. Double click the Android Studio icon to launch Android Studio.



1. In the **Welcome to Android Studio** dialog, click **New Project**.



The **New Project** window opens with a list of templates provided by Android Studio.



In Android Studio, a project template is an Android project that provides the blueprint for a certain type of app. Templates create the structure of the project and the files needed for Android Studio to build your project. The template that you choose provides starter code to get you going faster.

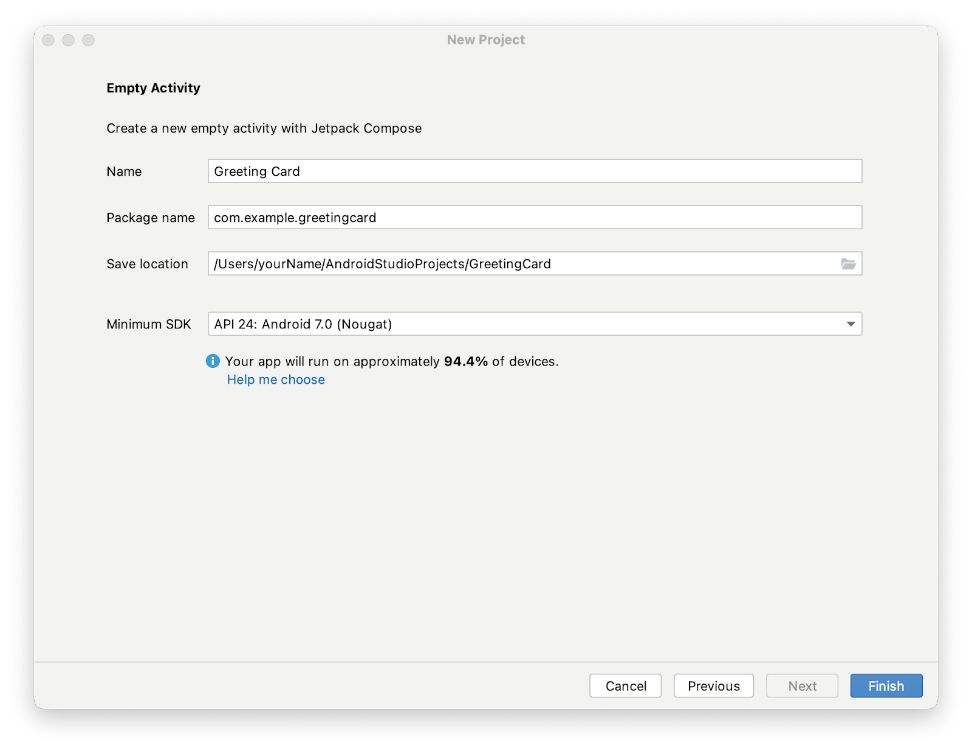
1. Make sure the **Phone and Tablet** tab is selected.
2. Click the **Empty Activity** template to select it as the template for your project. The **Empty Activity** template is the template to create a simple project that you can use to build a Compose app. It has a single screen and displays the text "Hello Android!".
3. Click **Next**. The **New Project** dialog opens. This has some fields to configure your project.
4. Configure your project as follows:

The **Name** field is used to enter the name of your project, for this codelab type "Greeting Card".

Leave the **Package name** field as is. This is how your files will be organized in the file structure. In this case, the package name will be com.example.greetingcard.

Leave the **Save location** field as is. It contains the location where all the files related to your project are saved. Take a note of where that is on your computer so that you can find your files.

Select **API 24: Android 7.0 (Nougat)** from the menu in the **Minimum SDK** field. [Minimum SDK](https://developer.android.com/guide/topics/manifest/uses-sdk-element) indicates the minimum version of Android that your app can run on.



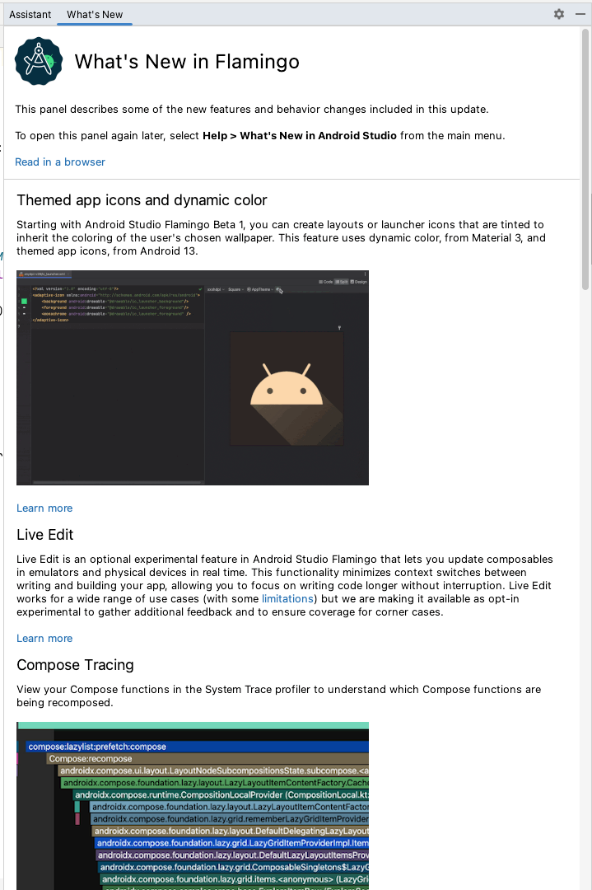
1. Click **Finish**. This may take a while - this is a great time to get a cup of tea! While Android Studio is setting up, a progress bar and message indicates whether Android Studio is still setting up your project. It may look like this:

This image shows a progress bar spinning and the text reads, 

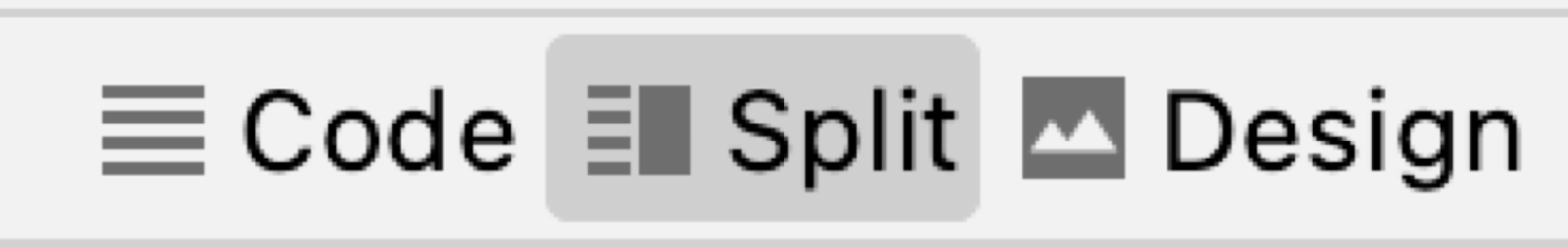
A message that looks similar to this informs you when the project set up is created.

This image shows a Gradle sync message that reads, 

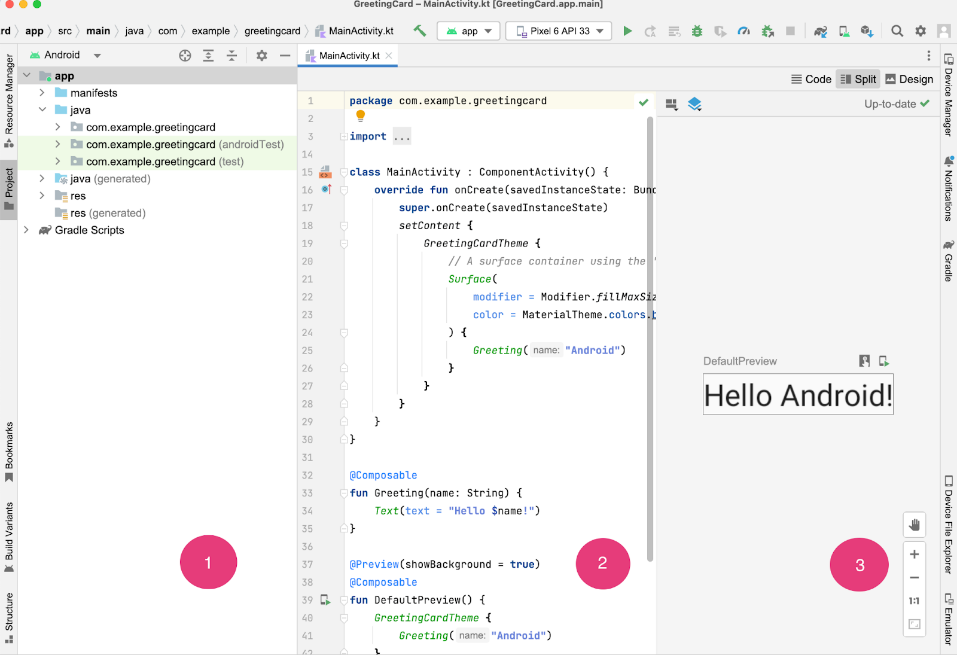
1. You may see a **What's New** pane which contains updates on new features in Android Studio. Close it for now.



1. Click **Split** on the top right of Android Studio, this allows you to view both code and design. You can also click **Code** to view code only or click **Design** to view design only.

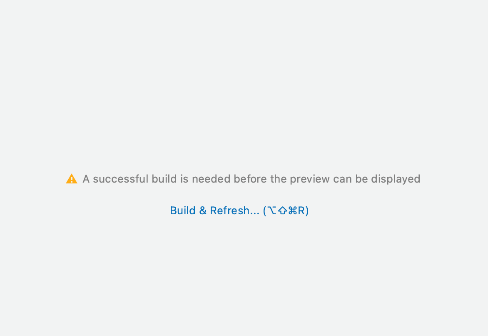


After pressing **Split** you should see three areas:

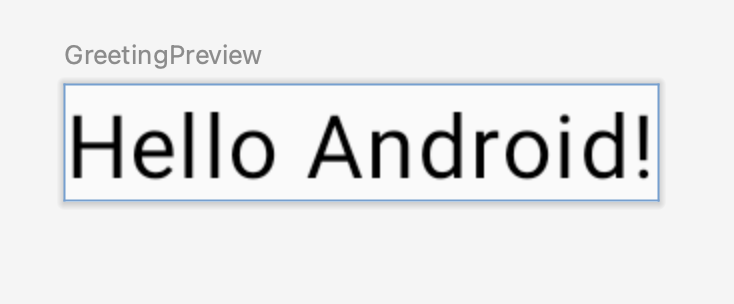


* The **Project** view (1) shows the files and folders of your project
* The **Code** view (2) is where you edit code
* The **Design** view (3) is where you preview what your app looks like

In the **Design** view, you will see a blank pane with this text:



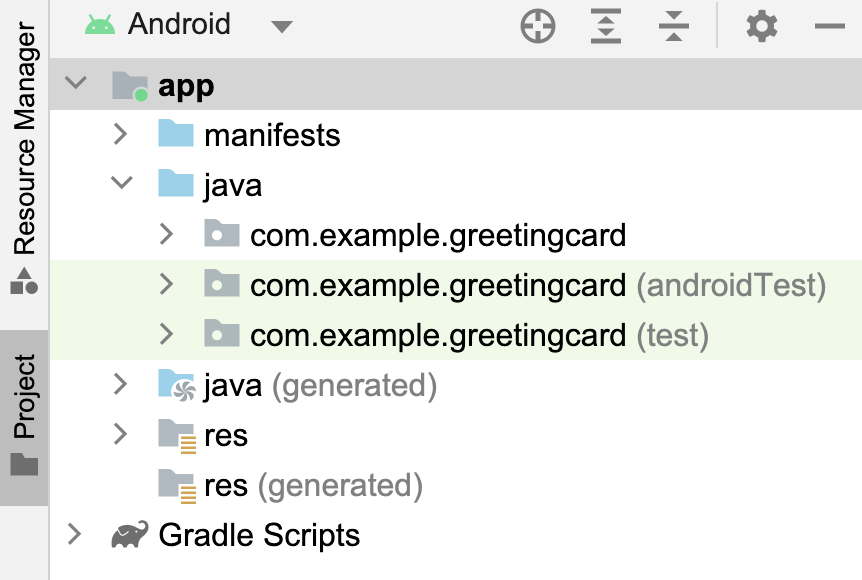
1. Click **Build & Refresh**. It may take a while to build but when it is done the preview shows a text box that says "**Hello Android!**". Empty Compose activity contains all the code necessary to create this app.



## 3. Find project files

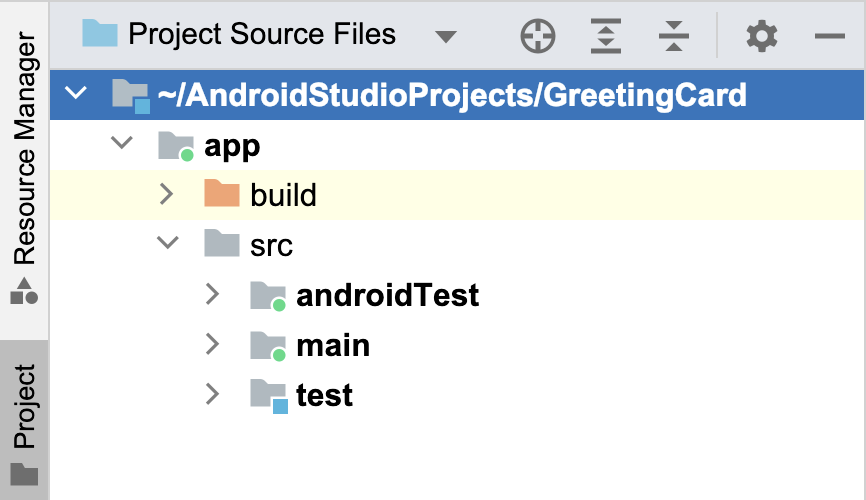
In this section you will continue to explore Android Studio by becoming familiar with the file structure.

1. In Android Studio, take a look at the **Project** tab. The **Project** tab shows the files and folders of your project. When you were setting up your project the package name was **com.example.greetingcard**. You can see that package right here in the **Project** tab. A package is basically a folder where code is located. Android Studio organizes the project in a directory structure made up of set of packages.
2. If necessary, select **Android** from the drop-down menu in the **Project** tab.



This is the standard view and organization of files that you use. It's useful when you write code for your project because you can easily access the files you will be working on in your app. However, if you look at the files in a file browser, such as Finder or Windows Explorer, the file hierarchy is organized very differently.

1. Select **Project Source Files** from the drop-down menu. You can now browse the files in the same way as in any file browser.



1. Select **Android** again to switch back to the previous view. You use the **Android** view for this course. If your file structure ever looks strange, check to make sure you're still in **Android** view.

## 4. Update the text

Now that you have gotten to know Android Studio, it's time to start making your greeting card!

Look at the **Code** view of the MainActivity.kt file. Notice there are some automatically generated functions in this code, specifically the onCreate() and the setContent() functions.

**Note:** Remember that a function is a segment of a program that performs a specific task.

class MainActivity : ComponentActivity() {  
   override fun onCreate(savedInstanceState: Bundle?) {  
       super.onCreate(savedInstanceState)  
       setContent {  
           GreetingCardTheme {  
               // A surface container using the 'background' color from the theme  
               Surface(  
                   modifier = Modifier.fillMaxSize(),  
                   color = MaterialTheme.colors.background  
               ) {  
                   Greeting("Android")  
               }  
           }  
       }  
   }  
}

The onCreate() function is the entry point to this Android app and calls other functions to build the user interface. In Kotlin programs, the main() function is the entry point/starting point of execution. In Android apps, the onCreate() function fills that role.

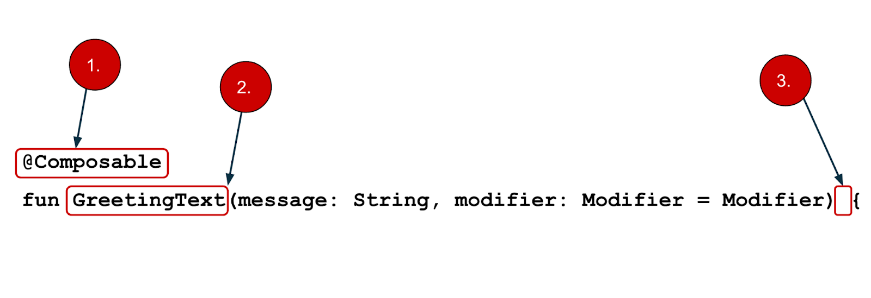
The [setContent()](https://developer.android.com/reference/kotlin/androidx/compose/ui/platform/ComposeView" \l "setContent(kotlin.Function0)" \t "_blank) function within the onCreate() function is used to define your layout through composable functions. All functions marked with the @Composable annotation can be called from the setContent() function or from other Composable functions. The annotation tells the Kotlin compiler that this function is used by Jetpack Compose to generate the UI.

**Note:** The compiler takes the Kotlin code you wrote, looks at it line by line, and translates it into something that the computer can understand. This process is called compiling your code.

Next, look at the Greeting() function. The Greeting() function is a Composable function, notice the @Composable annotation above it. This Composable function takes some input and generates what's shown on the screen.

@Composable  
fun Greeting(name: String, modifier: Modifier = Modifier) {  
    Text(text = "Hello $name!")  
}

You've learned about functions before (if you need a refresher, visit the [CreateandusefunctionsinKotlin](https://developer.android.com/codelabs/basic-android-kotlin-compose-functions" \t "_blank) codelab), but there are a few differences with composable functions.



* You add the @Composable annotation before the function.
* @Composable function names are capitalized.
* @Composable functions can't return anything.

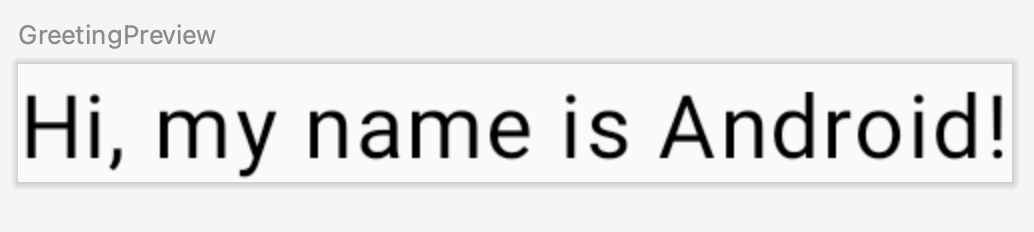
@Composable  
fun Greeting(name: String, modifier: Modifier = Modifier) {  
    Text(text = "Hello $name!")  
}

Right now the Greeting() function takes in a name and displays Hello to that person.

1. Update the Greeting() function to introduce yourself instead of saying "Hello":

@Composable  
fun Greeting(name: String, modifier: Modifier = Modifier) {  
    Text(text = "Hi, my name is $name!")  
}

1. Android should automatically update the preview.



Great! You changed the text, but it introduces you as Android, which is probably not your name. Next, you will personalize it to introduce you with your name!

The GreetingPreview() function is a cool feature that lets you see what your composable looks like without having to build your entire app. To enable a preview of a composable, annotate it with @Composable and @Preview. The @Preview annotation tells Android Studio that this composable should be shown in the design view of this file.

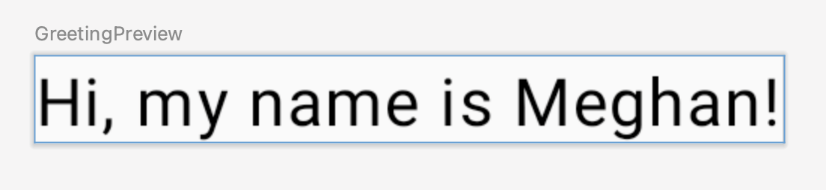
As you can see, the @Preview annotation takes in a parameter called showBackground. If showBackground is set to **true**, it will add a background to your composable preview.

Since Android Studio by default uses a light theme for the editor, it can be hard to see the difference between showBackground = true and showBackground = false. However, this is an example of what the difference looks like. Notice the white background on the image set to true.

|  |
| --- |
| showBackground = true |
| showBackground = false |

1. Update the GreetingPreview() function with your name. Then rebuild and check out your personalized greeting card!

@Preview(showBackground = true)  
@Composable  
fun GreetingPreview() {  
    GreetingCardTheme {  
        Greeting("Meghan")  
   }  
}

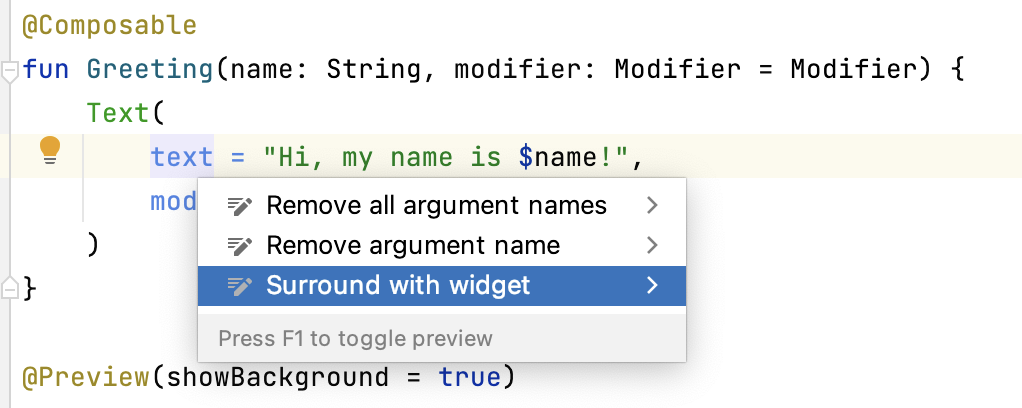


## 5. Change the background color

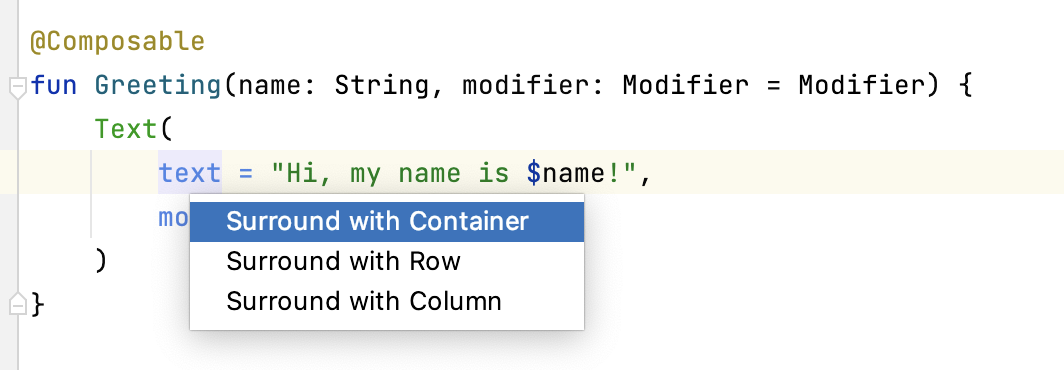
Now you have the introduction text, but it's a little boring! In this section, you learn to change the background color.

To set a different background color for your introduction, you'll need to surround your text with a [Surface](https://developer.android.com/reference/kotlin/androidx/compose/material/package-summary#Surface(androidx.compose.ui.Modifier,androidx.compose.ui.graphics.Shape,androidx.compose.ui.graphics.Color,androidx.compose.ui.graphics.Color,androidx.compose.foundation.BorderStroke,androidx.compose.ui.unit.Dp,kotlin.Function0)). A Surface is a container that represents a section of UI where you can alter the appearance, such as the background color or border.

1. To surround the text with a Surface, highlight the line of text, press (Alt+Enter for Windows or Option+Enter on Mac), and then select **Surround with widget**.



1. Choose **Surround with Container**.



The default container it will give you is Box, but you can change this to another container type. You will learn about Box layout later in the course.



1. Delete Box and type Surface() instead.

@Composable  
fun Greeting(name: String, modifier: Modifier = Modifier) {  
   Surface() {  
       Text(  
           text = "Hi, my name is $name!",  
           modifier = modifier  
       )  
   }  
}

1. To the Surface container add a color parameter, set it to Color.

@Composable  
fun Greeting(name: String, modifier: Modifier = Modifier) {  
   Surface(color = Color) {  
       Text(  
           text = "Hi, my name is $name!",  
           modifier = modifier  
       )  
   }  
}

1. When you type Color you may notice that it is red, which means Android Studio is not able to resolve this. To solve this scroll to the top of the file where it says import and press the three buttons.



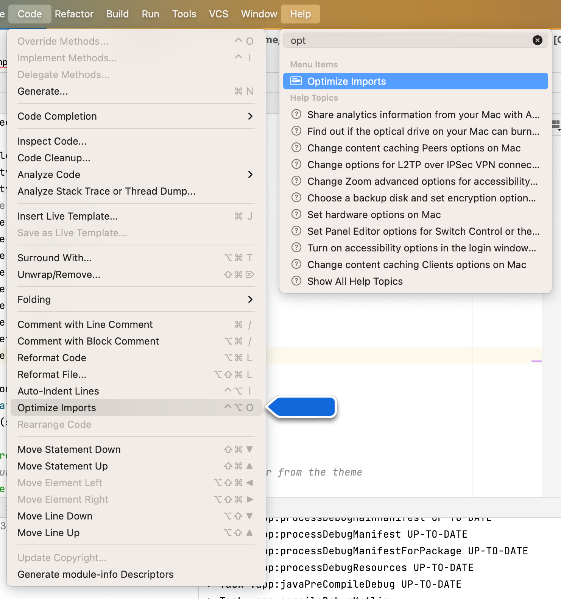
1. Add this statement to the bottom of the list of imports.

import androidx.compose.ui.graphics.Color

The full list of imports will look similar to this.

import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.layout.Box  
import androidx.compose.foundation.layout.fillMaxSize  
import androidx.compose.material3.MaterialTheme  
import androidx.compose.material3.Surface  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.tooling.preview.Preview  
import com.example.greetingcard.ui.theme.GreetingCardTheme  
import androidx.compose.ui.graphics.Color

1. In your code, the best practice is to keep your imports listed alphabetically and remove unused imports. To do this press **Help** on the top toolbar, type in **optimize imports**, and click on **Optimize Imports**.



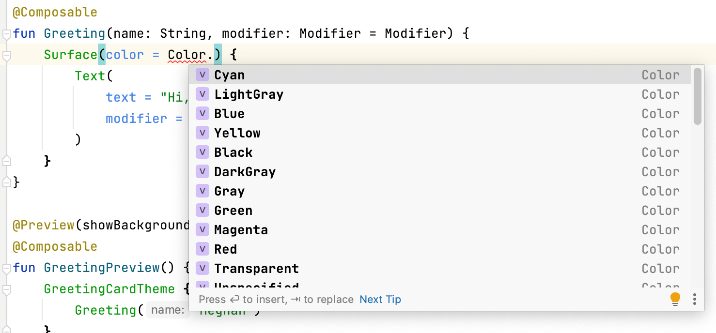
You could open the **Optimize Imports** directly from the menu: **Code** > **Optimize Imports.** Using Help's search option will help you locate a menu item if you don't remember where it is.

The full list of imports will now look like this:

import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.layout.fillMaxSize  
import androidx.compose.material3.MaterialTheme  
import androidx.compose.material3.Surface  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.graphics.Color  
import androidx.compose.ui.tooling.preview.Preview  
import com.example.greetingcard.ui.theme.GreetingCardTheme

1. Notice that the Color that you typed in the Surface parentheses has switched from being red to being underlined in red. To fix that, add a period after it. You will see a pop-up showing different color options.

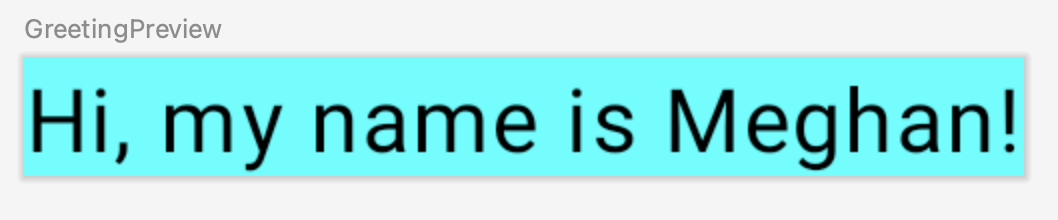
This is one of the cool features in Android Studio, it is intelligent and will help you out when it can. In this case it knows you are wanting to specify a color so it will suggest different colors.



1. Choose a color for your surface. This codelab uses **Cyan**, but you can choose your favorite!

@Composable  
fun Greeting(name: String, modifier: Modifier = Modifier) {  
   Surface(color = Color.Cyan) {  
       Text(  
           text = "Hi, my name is $name!",  
           modifier = modifier  
       )  
   }  
}

1. Notice the updated preview.



## 6. Add padding

Now your text has a background color, next you will add some space (padding) around the text.

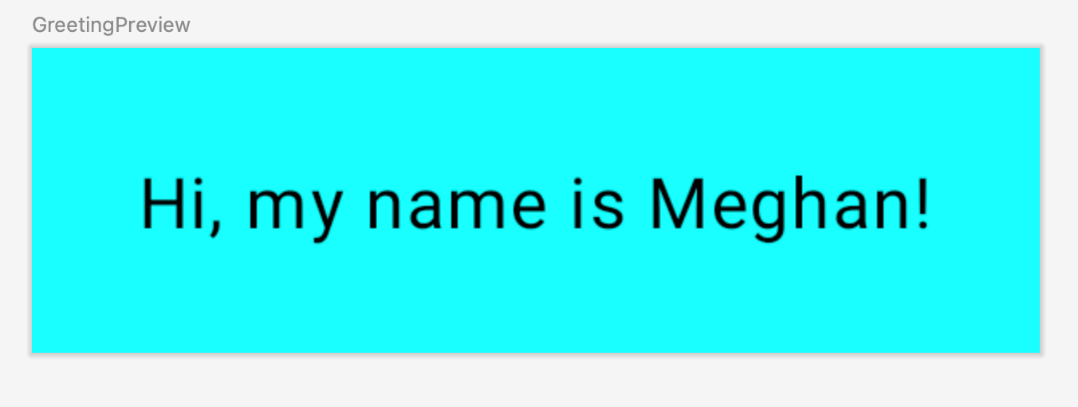
A [Modifier](https://developer.android.com/reference/kotlin/androidx/compose/ui/Modifier) is used to augment or decorate a composable. One modifier you can use is the padding modifier, which adds space around the element (in this case, adding space around the text). This is accomplished by using the [Modifier.padding()](https://developer.android.com/reference/kotlin/androidx/compose/ui/Modifier" \l "(androidx.compose.ui.Modifier).padding(androidx.compose.ui.unit.Dp)" \t "_blank) function.

Every composable should have an optional parameter of the type Modifier. This should be the first optional parameter.

1. Add a padding to the modifier with a size of 24.dp.

**Note:** You learn more about density-independent pixels (**dp**) in the next pathway, but refer to [Layout – Material Design 3](https://m3.material.io/foundations/layout/understanding-layout/spacing#abccd6ce-1092-4ad0-9351-de75aeae0edf) article if you want to read more now.

@Composable  
fun Greeting(name: String, modifier: Modifier = Modifier) {  
   Surface(color = Color.Cyan) {  
       Text(  
           text = "Hi, my name is $name!",  
           modifier = modifier.padding(24.dp)  
       )  
   }  
}



1. Add these imports to the import statement section.

Make sure to use **Optimize Imports** to alphabetize the new imports.

import androidx.compose.ui.unit.dp  
import androidx.compose.foundation.layout.padding

Well congratulations - you built your first Android app in Compose! This is a pretty huge accomplishment. Take some time to play around with different colors and text, make it your own!

## 7. Review the solution code

### Code snippet for review

package com.example.greetingcard  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.ui.graphics.Color  
import androidx.compose.foundation.layout.fillMaxSize  
import androidx.compose.foundation.layout.padding  
import androidx.compose.material3.MaterialTheme  
import androidx.compose.material3.Surface  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.tooling.preview.Preview  
import androidx.compose.ui.unit.dp  
import com.example.greetingcard.ui.theme.GreetingCardTheme  
  
class MainActivity : ComponentActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContent {  
            GreetingCardTheme {  
                // A surface container using the 'background' color from the theme  
                Surface(  
                    modifier = Modifier.fillMaxSize(),  
                    color = MaterialTheme.colorScheme.background  
                ) {  
                    Greeting("Android")  
                }  
            }  
        }  
    }  
}  
  
@Composable  
fun Greeting(name: String, modifier: Modifier = Modifier) {  
    Surface(color = Color.Cyan) {  
        Text(text = "Hi, my name is $name!", modifier = modifier.padding(24.dp))  
    }  
}  
  
@Preview(showBackground = true)  
@Composable  
fun GreetingPreview() {  
    GreetingCardTheme {  
        Greeting("Meghan")  
    }  
}

1. **Build Lists and Add Theme**

* Use data classes, functions, and collections in Kotlin.
* Create a scrollable list in an app that displays both text and images.
* Add click listeners to interact with list items.
* Add an app bar to the app and modify the app theme.
* Use Material Design to build modern and intuitive user interfaces, using colors, shapes and typography.

1. **Navigation and App Architecture**

* Explain activities and their lifecycles.
* Understand Modern Android architecture.
* Use StateFlow and UDF pattern to work with state and events.
* Add a ViewModel to save data and state.
* Set up and use the Navigation component with Compose.
* Understand what responsive UI is.
* Use window class sizes to build for different screen sizes.
* Add a navigation drawer to an app.

1. **Connect to the Internet**

* Describe the basics of concurrency and how to use coroutines in an Android app.
* Define and understand the data layer in Modern Android app architecture.
* Implement a repository to centralize data access

.

* Use Retrofit to retrieve data from a remote server.
* Load and display images using the Coil library.
* Implement dependency injection to decouple the classes, making it easier to test, maintain, and scale

the app.

1. **Data Persistence**

* Learn the basics of SQL to insert, update, and delete data from a SQLite database.
* Use the Room library to add a database to an Android app.
* Use Database Inspector to test and debug database issues.
* Use Preference DataStore to store user preferences.

1. **WorkManager**

* Define long running tasks that need to run in background work.
* Add WorkManager to an Android app.
* Create a Worker object and enqueue work.
* Create constraints on WorkRequests.
* Use the Background Task Inspector to inspect and debug WorkManager

1. **Compose with Views**

* Understand the View-based UI toolkit and build app UI using XML.
* Add a composable in an app built with Views.
* Add Navigation component to the app and use it to navigate between fragments.
* Use AndroidView to display views.
* Add existing View-based UI components in a Compose app.

1. **Project Description**

**Project: Android News App**

**Abstract:**

As world’s technology is rapidly growing we has fast connection and network to instantly connect to

other person. Day to day use in mobile, tablets and laptop is increasing, most of the people already have this

facilities. In this fast and information oriented world we need to stay updated with every incidents and news

too. This News app is android mobile application where user have access to latest news from 120+

newspapers from 50+ countries. The main focus of this application is to connect news articles from all around

the world and deliver it to user as fast as possible in best visualize way.

**INTRODUCTION:**

Android provides simple application structure and requires Java and Mark-up languages

knowledge to work with. Such as, an discrete movement delivers a solitary screen for a user interface and a

service whole completes work in the contextual [1]. We can work on different module separately and can

combine at the end, we can also add future modules easily afterwards. API (Application Programming

Interface) which is an intermediate interface between different applications. It provides automation,

immediacy, adaption and personalization. News API provides us the source of news articles from many

different sources at one place and updates it. To expand the sources old fashioned Admin panel can be used

where writers will fill the gap of API. In 2014, a design language has been created by google named Material

Design which is based on “cards” uses grind based layouts, responsive animation, padding and depth effects

like shadow to create an responsive, attractive and easy user interface. With the use of different libraries and

material design it is possible to use attractive UI.

**RELATED WORK:**

Native news apps are expensive and difficult to maintain. Native Publishers like BBC

News or NY Times uses their own writers to manage articles and manage it. Many native newspaper are

divided because of this which causes in lack of resources from one side. Android structure provides great

capability with frameworks, libraries and APIs, with the help of it we can provide better user experience and

combine this sources at one place while maintaining integrity of its owner.

**MODULES:**

**User Interface**: One of the factors in successful news app development is visualization of news and its feature with user. For the development of an android app material design is very useful and provides smooth experience with custom layout, views and animations. For this news app user should be able to select from different categories, countries and newspaper. Short News as list view with header, little description and image before showing full article can be helpful to user to determine what type of news they are looking for. View Holder can be used for this list view for better and fast experience. Library like Picasso can be used for better image handling. This User interface will be connected to API and Admin Panel database which will give full article in form of web view of that article. Because of this structure the integrity of writer of that article will not be in harm.

**API:** News API has been used for collecting different news sources at one spot. On sending request it will give response in JSON format which contains source id, title, description, image URL, article URL, author, time etc. We need to handle and parse this JSON into string format which is our required format.

**Admin Panel:** This module of app controls the User and Writers logins from database. Writers can add news, update and delete from its database as per required. Writers will only has access to admin panel while Main Admin will has access to database as well. Main Admin can add Users, Writers, and News. He can also approve, update and delete it. Using this approach we can create network in local areas connect by writers and local admins which will provide news at local level and we can also implement location feature which will update local news of different location or city.

**AVAILABLE FEATURES**

**Global Support**: Different type of newspaper will be available from all around the world in different languages with this user will be able to get news from all around the world.

**Short News:** News will be displayed in short format with title, image and little description in list view. It will help user to access required news faster.

**Search Option**: User will be able to search from not only one source but many different sources available within API.

**Favorites / Offline Reading**: News can be added as favourites which will automatically will be saved for offline reading.

**Sharing:** User will be able to share news easily on social media.

**EXPERIMENTAL STUDY AND RESULT :**

User was allowed to use this application in his smartphone and screenshots were taken as a result for this study. First User need to Sign Up in order to access the application which provides security for this application. Also predicted user error handling with pop-up messaging was done before this experiment like entering invalid data in fields, not selecting a field before clicking on action button etc.

1. **Implementation of Project**

**Below are the implementation steps for the news app Android application development project:**

**Step 1: Set Up Project**

Create a new Android Studio project.

Choose an appropriate project template and configure the project settings.

**Step 2: Dependencies**

Add the Retrofit dependencies to the build.gradle file as mentioned in the previous response.

Sync the project to fetch the new dependencies.

**Step 3: Design UI**

Create the layout for activity\_main.xml containing a RecyclerView.

Create the layout for news\_item.xml to represent each item in the RecyclerView.

**Step 4: Create Model**

Create the NewsModel.kt data class to represent the structure of a news article.

Create the NewsResponse.kt data class to represent the structure of the API response.

**Step 5: Create Adapter**

Create the NewsAdapter.kt class to bind the data to the RecyclerView.

Implement the necessary methods like onCreateViewHolder and onBindViewHolder.

**Step 6: Create API Service Interface**

Create the NewsApiService.kt interface using Retrofit annotations to define the API endpoints.

**Step 7: Create API Client**

Create the NewsApiClient.kt object to set up Retrofit and provide an instance of the API service.

**Step 8: Implement MainActivity**

Open MainActivity.kt.

Set up the RecyclerView and adapter in onCreate.

Create a function to fetch news data using Retrofit in a coroutine.

**Handle success and error cases in fetching news data.**

**Step 9: Request API Key**

Visit the News API website and sign up for an API key.

Replace "YOUR\_API\_KEY" in NewsApiService.kt with your actual API key.

**Step 10: Add Internet Permission**

Open the AndroidManifest.xml file.

Add the internet permission within the <uses-permission> tag.

**Step 11: Test the App**

Run the app on an emulator or physical device.

Ensure that the news articles are displayed correctly.

**Step 12: Enhance Features**

Implement error handling for network requests.

Add loading indicators to provide feedback to the user.

Implement pull-to-refresh functionality.

Consider adding details view for each news article.

Apply any additional features or enhancements based on your project requirements**.**

**Step 13: Code Cleanup**

Refactor and organize code for better readability.

Remove any unused or unnecessary code.

Add comments and documentation to explain critical sections of the code.

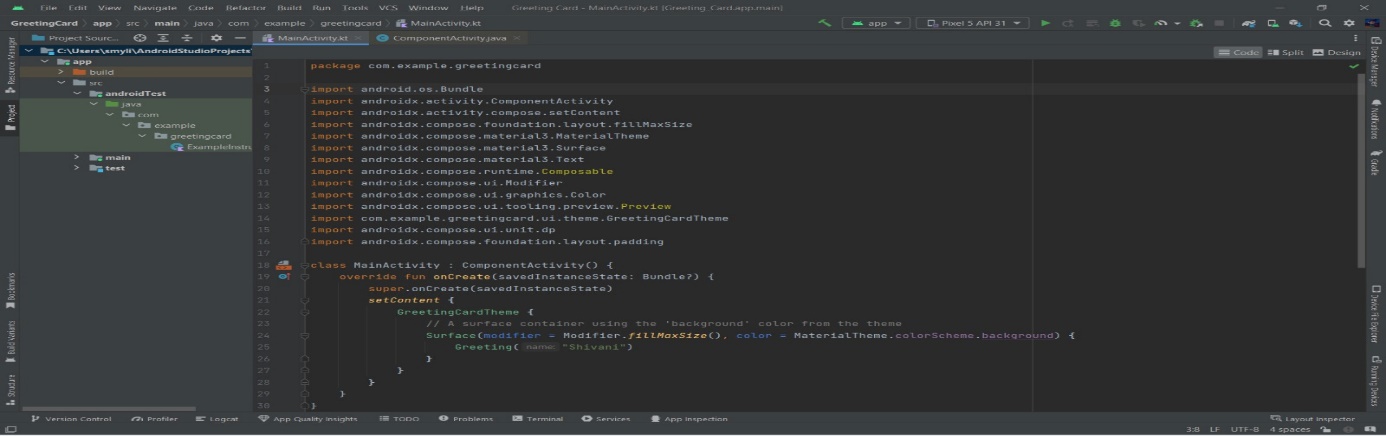
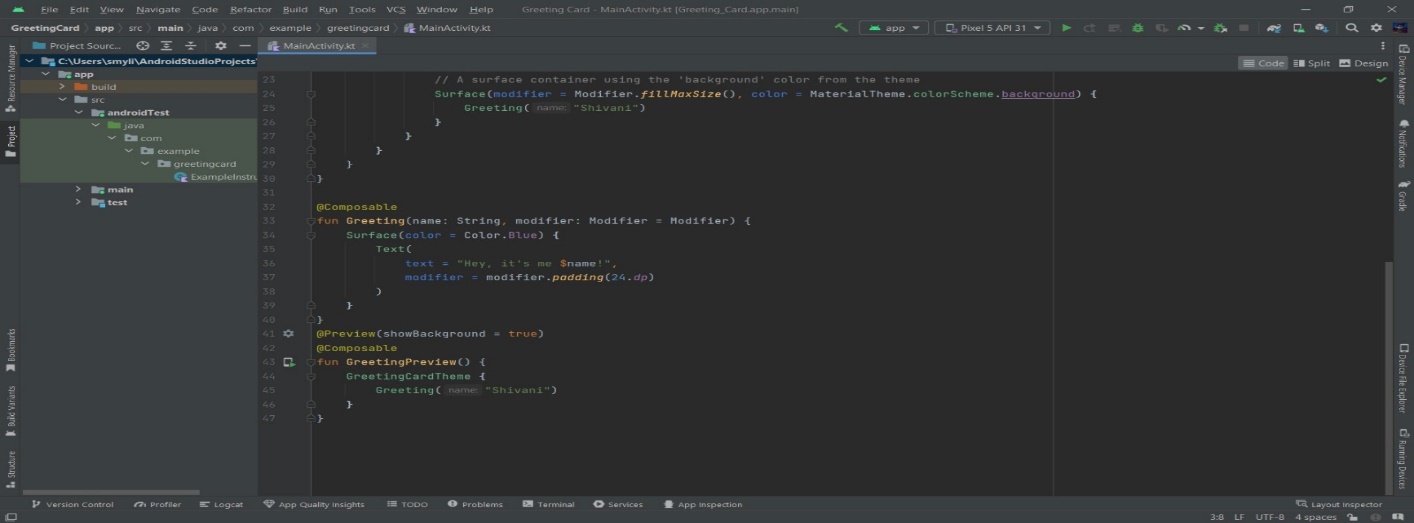
**Step 14: Testing**

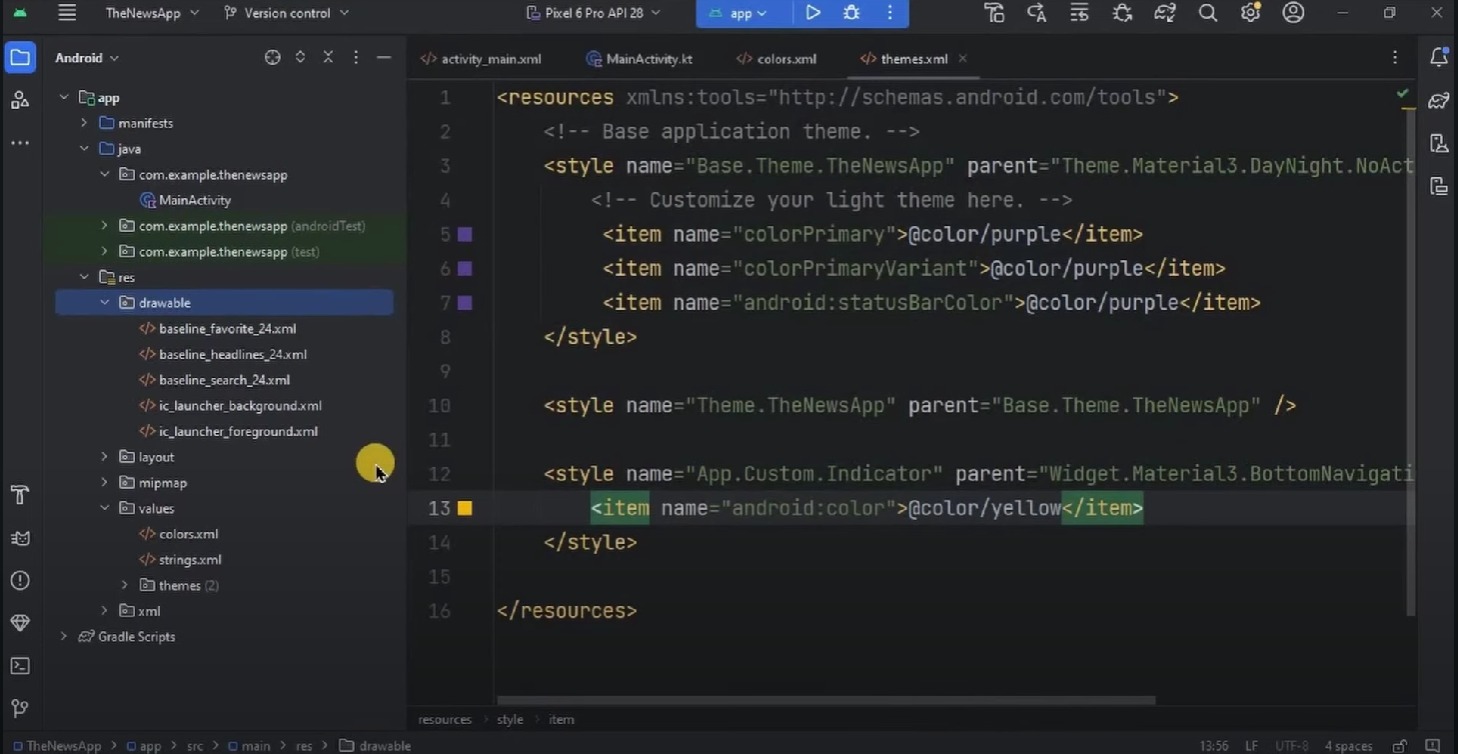
Write unit tests for critical functions using frameworks like JUnit or Kotlin Test.

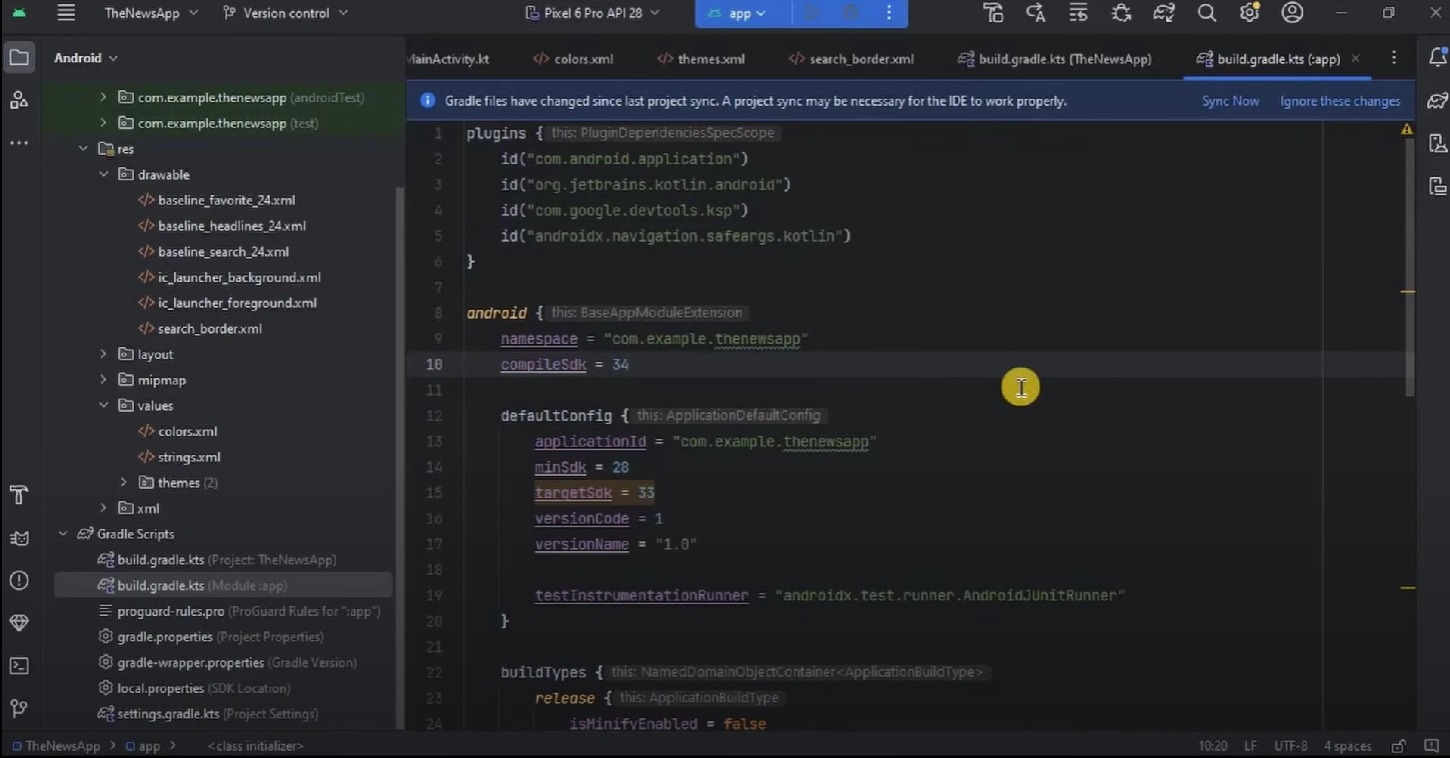
Conduct manual testing to ensure the app works as expected.

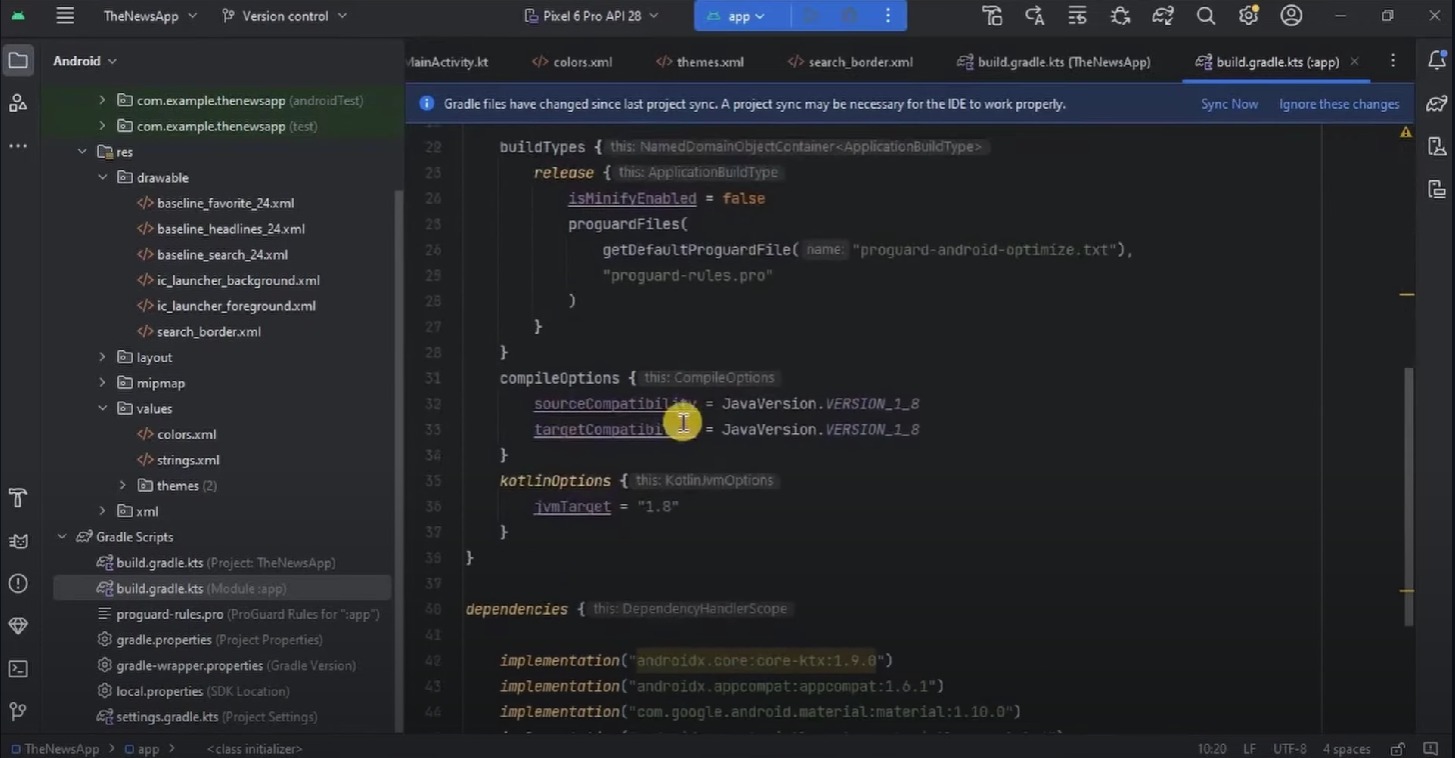
Step 15: Publish

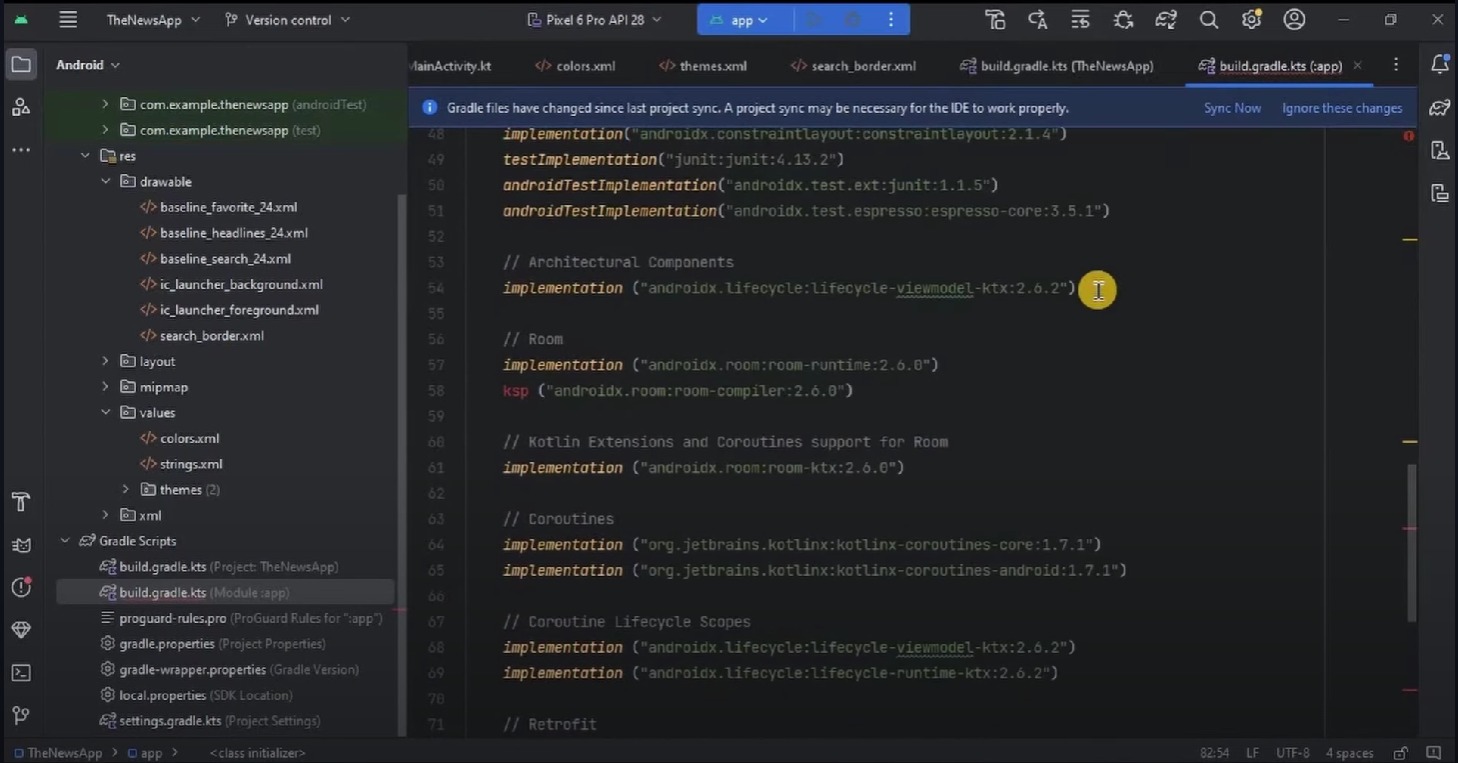
If ready, prepare the app for release.

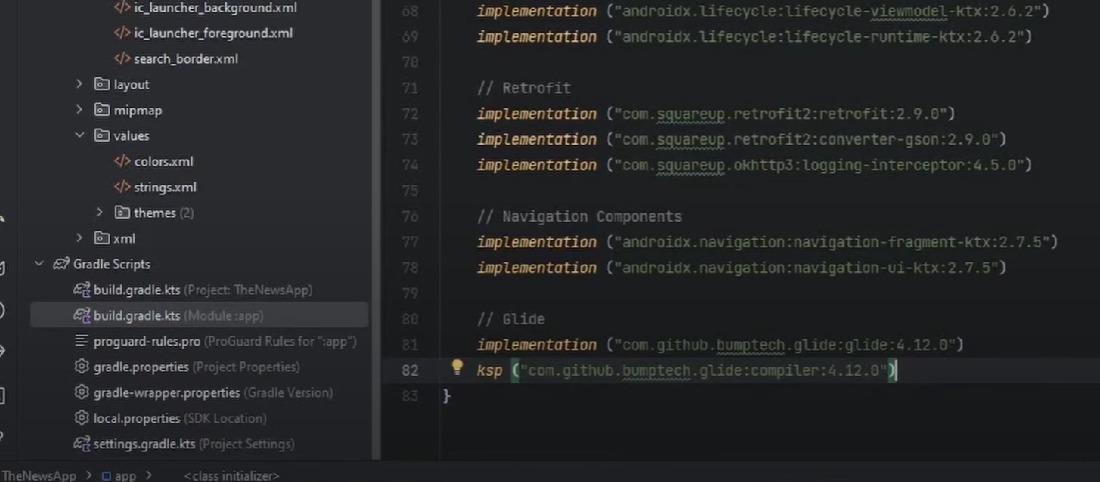
1. **Screenshots:**

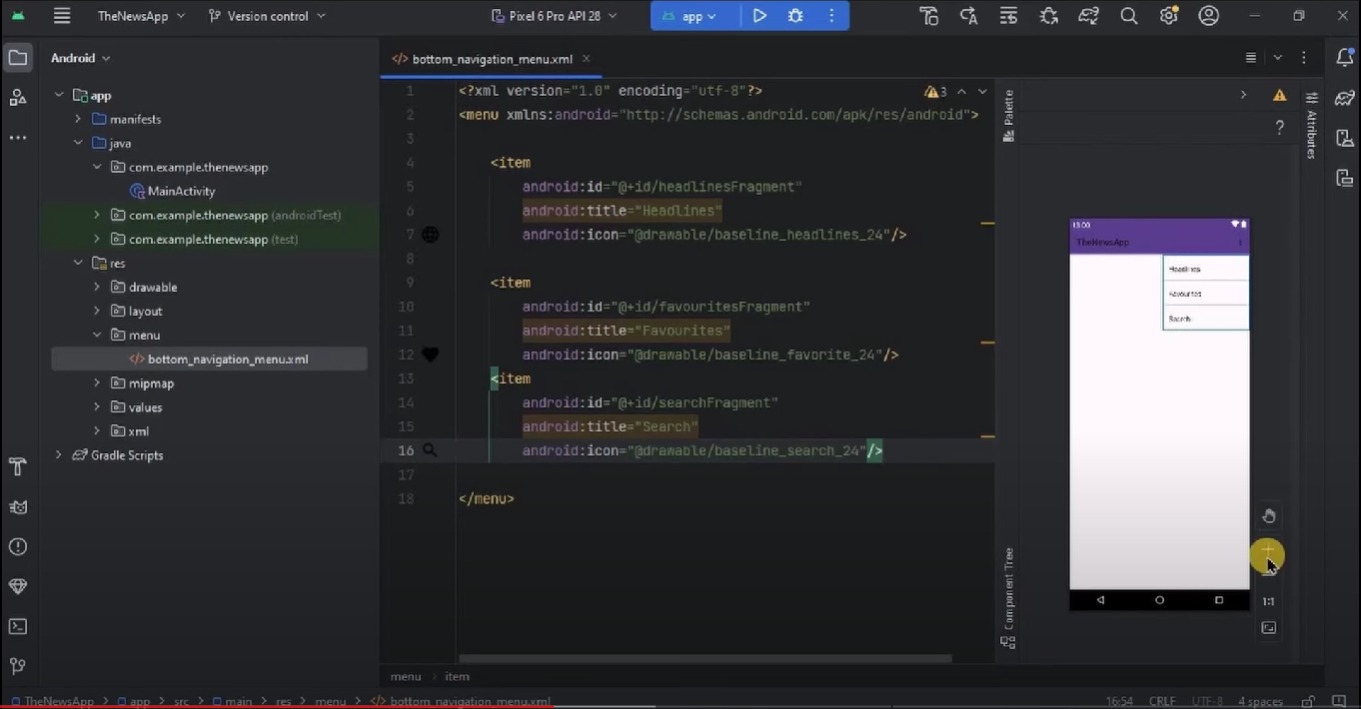
****

****

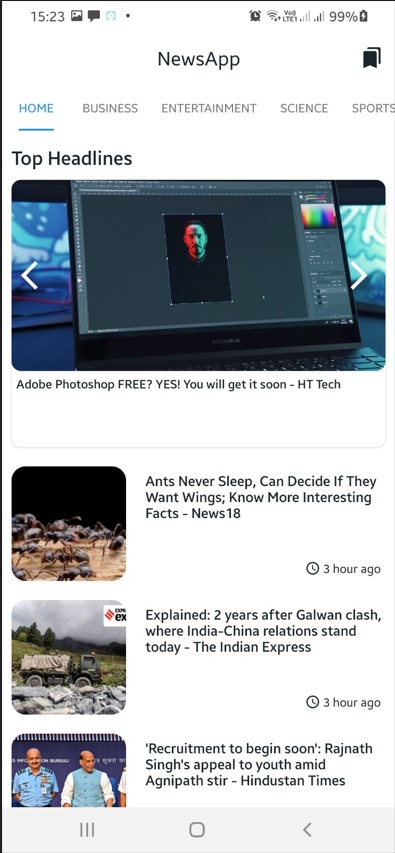
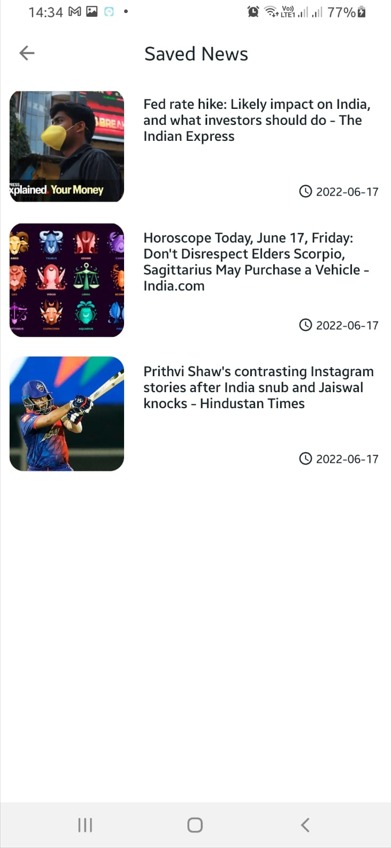
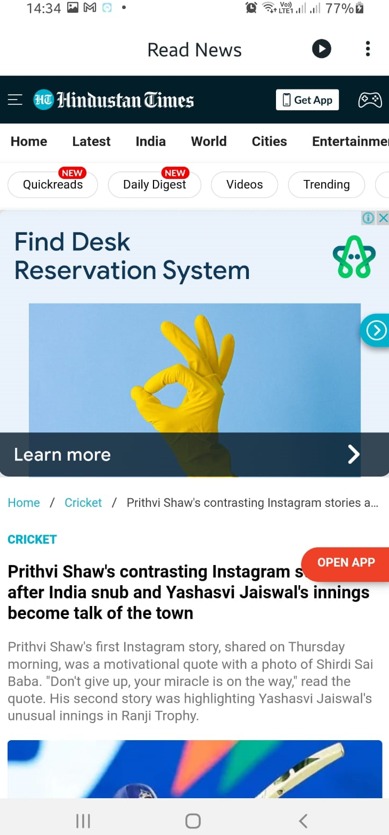
****







**Output**:

1. **Reflection on Internship**

My internship in Android Application Development using Kotlin has been a journey of discovery and growth. From the outset, I was immersed in diverse datasets, challenging me to apply theoretical knowledge to real-world scenarios. The hands-on experience with tools has been instrumental in expanding my technical proficiency. Through collaborative projects, I've learned to navigate the complexities of team dynamics and appreciate the synergy required for effective data-driven decision-making. One key takeaway has been the significance of adaptability in the face of evolving data landscapes. The rapid pace of technological advancements became evident, prompting me to stay abreast of industry trends and continuously upskill. Additionally, I've developed a keen awareness of the ethical considerations surrounding data analytics, emphasizing the importance of responsible and transparent practices.

Moreover, this internship has underscored the vital role of effective communication in the analytics process. Presenting findings to both technical and non-technical stakeholders challenged me to convey complex insights in a clear and compelling manner, bridging the gap between data and actionable strategies.

In retrospect, this internship has not only refined my technical skills but has also cultivated a holistic understanding of the multifaceted nature of data analytics. As I move forward in my career, I carry with me a sense of accomplishment, equipped with the practical insights and lessons gained during this enriching experience.

## CONCLUSION

In conclusion, my internship in Android Application Development using Kotlin has been a transformative experience, providing me with hands-on exposure to real-world data challenges. I've honed my analytical and technical skills, gained proficiency in tools like Sklearn, Linear Regression, Logistic Regression, and collaborated effectively with cross- functional teams. This internship has not only deepened my understanding of data analysis but also equipped me with practical insights that will undoubtedly contribute to my future endeavors in the field. I am grateful for the opportunities for growth and learning, and I look forward to applying these skills in future roles. Furthermore, this internship has reinforced the importance of clear communication in translating complex findings into actionable insights. I've had the opportunity to present my analyses to stakeholders, refining my ability to convey technical information in a digestible manner. The constructive feedback received has been invaluable, shaping my communication skills and enhancing my overall professional development. As I reflect on this experience, I am confident that the lessons learned and the relationships built will be integral to my continued growth in the dynamic and evolving field of data analytics.