ANDROID ASSIGNMENT

Research

1. What is Android? Who created it? What are Android Apps?

**Android** is an open-source operating system based on Linux with a Java programming interface for mobile devices such as Smartphone (Touch Screen Devices who supports Android OS) as well for Tablets too.

Android was developed by the **Open Handset Alliance** (OHA), which is led by Google. The **Open Handset Alliance** (OHA) is a consortium of multiple companies like Samsung, Sony, Intel and many more to provide services and deploy handsets using the android platform.

All the source code for Android is available free on Git-Hub, Stack overflow, and many more websites. Google publishes most of the code under the Apache License version 2.0.

1. What is the software used in the development of Android Apps?

Android OS is created using Java, so knowledge about Java is always a benefit if you want to develop an app for Android.

Then, you will need to get Android SDK, which is Android’s basic app writing program like Android Studio or Eclipse, the Java Software Development Kit (JDK). The built-in features of Android SDK provide the base needed to build some great mobile applications and providing a great opportunity for developers and entrepreneurs.

The most common IDE for Android development is Android Studio, which comes directly from Google itself. The amazing thing about Android Studio is that it is designed specifically for Android app development services.

1. Which are the languages commonly used in the development of android apps? Which language does InstiApp use?

Android apps can be created using HTML, CSS, and JavaScript using the Adobe PhoneGap framework that is powered by Apache Cordova.

Java is the official language for Android App Development and consequently, it is the most used language as well.

Kotlin is a cross-platform programming language that may be used as an alternative to Java for Android App Development.

C++ can be used for Android App Development using the Android Native Development Kit(NDK).

Python can be used for Android App Development even though Android doesn’t support native Python development. This can be done using various tools that convert the Python apps into Android Packages that can run on Android devices.

INSTIAPP

The API Backend of the app, which is a RESTful API in Django. This project enforces very strict coding standards and testing routines, so you will essentially learn not only Python and Django, but software development as a whole.

A Progressive Web Application (PWA), written in Angular. You will learn fairly advanced TypeScript, HTML, CSS and JavaScript if you are working on this. It will also involve a lot of async programming, cache handling, performance optimization etc.

A native Android App, which is a separate project

1. What is the activity cycle of a basic Android application? Diagrams/flowcharts preferred.

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: [onCreate()](https://developer.android.com/reference/android/app/Activity#onCreate(android.os.Bundle)), [onStart()](https://developer.android.com/reference/android/app/Activity#onStart()), [onResume()](https://developer.android.com/reference/android/app/Activity#onResume()), [onPause()](https://developer.android.com/reference/android/app/Activity#onPause()), [onStop()](https://developer.android.com/reference/android/app/Activity#onStop()), and [onDestroy()](https://developer.android.com/reference/android/app/Activity" \l "onDestroy()). The system invokes each of these callbacks as an activity enters a new state.



1. What are 5 different UI elements in an android app? One example is a “TextView”.

In android **UI** or**input** controls are the interactive or View components that are used to design the user interface of an application. In android we have a wide variety of UI or input controls available, those are [TextView](https://www.tutlane.com/tutorial/android/android-textview-with-examples), [EditText](https://www.tutlane.com/tutorial/android/android-edittext-with-examples), [Buttons](https://www.tutlane.com/tutorial/android/android-button-with-examples), [Checkbox](https://www.tutlane.com/tutorial/android/android-checkbox-with-examples), [Progressbar](https://www.tutlane.com/tutorial/android/android-progressbar-with-examples" \o "Android ProgressBar Control with Examples" \t "_blank), [Spinners](https://www.tutlane.com/tutorial/android/android-spinner-dropdown-list-with-examples), etc.

Android UI Controls

There are number of UI controls provided by Android that allow you to build the graphical user interface for your app.

|  |  |
| --- | --- |
| Sr.No. | UI Control & Description |
| 1 | [TextView](https://www.tutorialspoint.com/android/android_textview_control.htm)  This control is used to display text to the user. |
| 2 | [EditText](https://www.tutorialspoint.com/android/android_edittext_control.htm)  EditText is a predefined subclass of TextView that includes rich editing capabilities. |
| 3 | [AutoCompleteTextView](https://www.tutorialspoint.com/android/android_autocompletetextview_control.htm)  The AutoCompleteTextView is a view that is similar to EditText, except that it shows a list of completion suggestions automatically while the user is typing. |
| 4 | [Button](https://www.tutorialspoint.com/android/android_button_control.htm)  A push-button that can be pressed, or clicked, by the user to perform an action. |
| 5 | [ImageButton](https://www.tutorialspoint.com/android/android_imagebutton_control.htm)  An ImageButton is an AbsoluteLayout which enables you to specify the exact location of its children. This shows a button with an image (instead of text) that can be pressed or clicked by the user. |
| 6 | [CheckBox](https://www.tutorialspoint.com/android/android_checkbox_control.htm)  An on/off switch that can be toggled by the user. You should use check box when presenting users with a group of selectable options that are not mutually exclusive. |
| 7 | [ToggleButton](https://www.tutorialspoint.com/android/android_togglebutton_control.htm)  An on/off button with a light indicator. |
| 8 | [RadioButton](https://www.tutorialspoint.com/android/android_radiobutton_control.htm)  The RadioButton has two states: either checked or unchecked. |
| 9 | [RadioGroup](https://www.tutorialspoint.com/android/android_radiogroup_control.htm)  A RadioGroup is used to group together one or more RadioButtons. |
| 10 | [ProgressBar](https://www.tutorialspoint.com/android/android_progressbar.htm)  The ProgressBar view provides visual feedback about some ongoing tasks, such as when you are performing a task in the background. |
| 11 | [Spinner](https://www.tutorialspoint.com/android/android_spinner_control.htm)  A drop-down list that allows users to select one value from a set. |
| 12 | [TimePicker](https://www.tutorialspoint.com/android/android_timepicker_control.htm)  The TimePicker view enables users to select a time of the day, in either 24-hour mode or AM/PM mode. |
| 13 | [DatePicker](https://www.tutorialspoint.com/android/android_datepicker_control.htm)  The DatePicker view enables users to select a date of the day. |

1. [BONUS]What are some of the salient features of those languages (part c)? How similar are they to C++?

Android apps can be created using HTML, CSS, and JavaScript using the Adobe PhoneGap framework that is powered by Apache Cordova. The PhoneGap framework basically allows the usage of web development skills to create hybrid apps that are shown through “WebView” but are packaged like an app.

While the Adobe PhoneGap framework is enough for basic tasks in the realm of Android App Development, it hardly requires much programming except for JavaScript. And since it needs a lot of work to even create a decent app, it is better to use the other languages in this list if you want to be called a true Android developer(Yes…That’s a thing!)

Java is the official language for Android App Development and consequently, it is the most used language as well. Many of the apps in the Play Store are built with Java and it is also the most supported language by Google. In addition to all this, Java has a great online community for support in case of any problems (And trust me, there will be problems!).

Kotlin is a cross-platform programming language that may be used as an alternative to Java for Android App Development. It has also been introduced as a secondary “official” Java language in 2017. Kotlin can inter-operate with Java and it runs on the Java Virtual Machine.

The only sizable difference is that Kotlin removes the superfluous features of Java such as null pointer exceptions. It also removes the necessity of ending every line with a semicolon. In short, Kotlin is much simpler for beginners to try as compared to Java and it can also be used as an “entry point” for Android App Development.

C++ can be used for Android App Development using the Android Native Development Kit(NDK). However, an app cannot be created totally using C++ and the NDK is used to implement parts of the app in C++ native code. This helps in using C++ code libraries for the app as required.

While C++ is useful for Android App Development in some cases, it is much more difficult to set up and is much less flexible. It may also lead to more bugs because of the increased complexity. So, it is better to use Java as compared to C++ as it does not provide enough gain to offset the efforts required.

Python can be used for Android App Development even though Android doesn’t support native Python development. This can be done using various tools that convert the Python apps into Android Packages that can run on Android devices.

**TASK**

1. Read about relative and linear layouts and how they are used to design the UI of Apps.

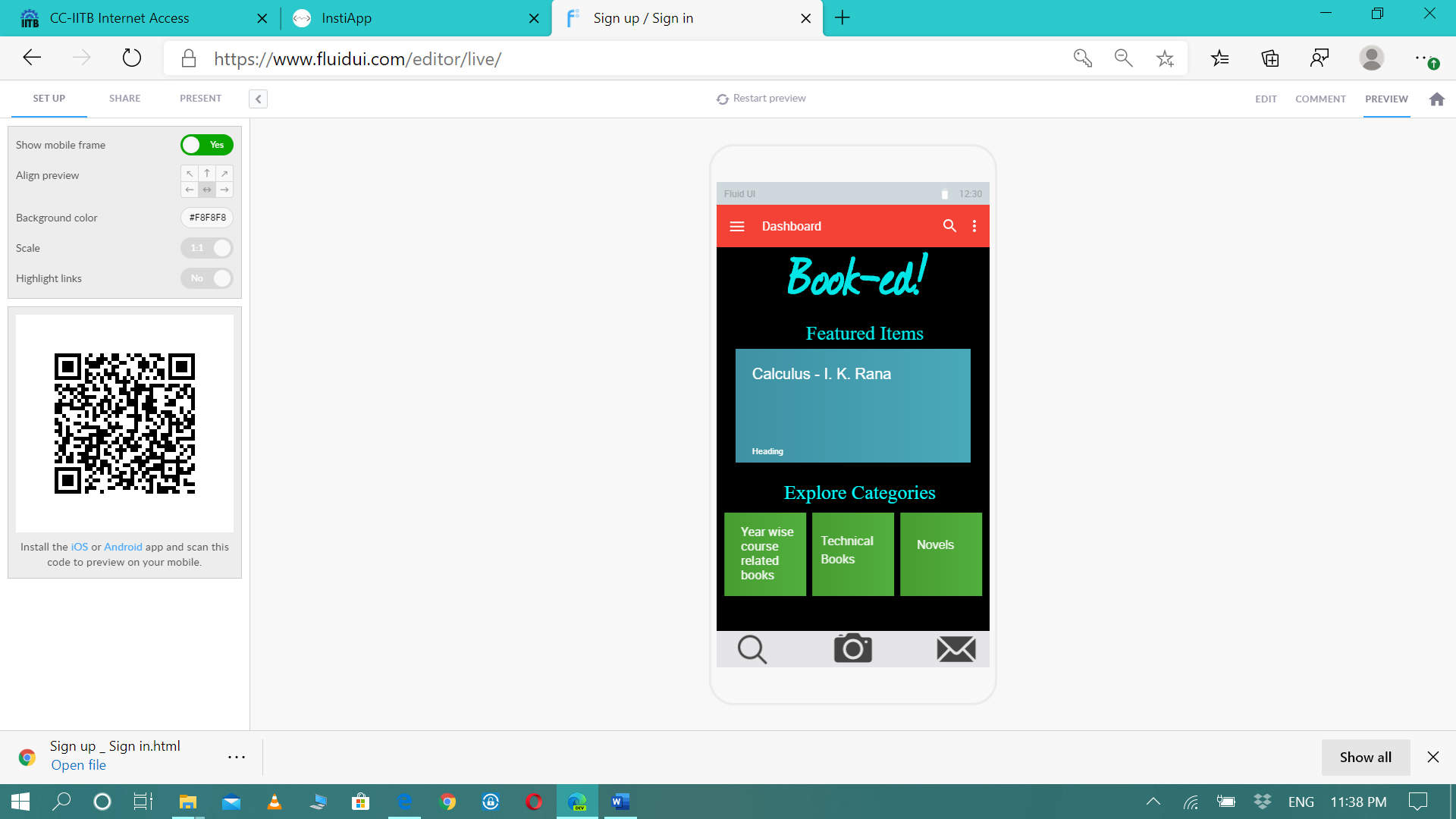
A layout defines the structure for a user interface in your app, such as in an [activity](https://developer.android.com/guide/components/activities). All elements in the layout are built using a hierarchy of [View](https://developer.android.com/reference/android/view/View) and [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup) objects. A [View](https://developer.android.com/reference/android/view/View) usually draws something the user can see and interact with. Whereas a [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup) is an invisible container that defines the layout structure for [View](https://developer.android.com/reference/android/view/View) and other [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup) objects

|  |  |
| --- | --- |
| Relative    A [RelativeLayout](https://developer.android.com/reference/android/widget/RelativeLayout) is a very powerful utility for designing a user interface because it can eliminate nested view groups and keep your layout hierarchy flat, which improves performance. If you find yourself using several nested LinearLayout groups, you may be able to replace them with a single RelativeLayout.  RelativeLayout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center). | Linear    All children of a [LinearLayout](https://developer.android.com/reference/android/widget/LinearLayout) are stacked one after the other, so a vertical list will only have one child per row, no matter how wide they are, and a horizontal list will only be one row high (the height of the tallest child, plus padding). A LinearLayout respects margins between children and the gravity (right, center, or left alignment) of each child. |

1. Now suppose you want to design the landing screen / dashboard of Book-ed!, what do you  think should be the various features of that screen?

* Title – Book-ed!
* what are you looking for = search bar
* inbox
* Featured items
* Year wise course related books
* Explore categories
  + Technical books
  + Non-technical books
  + novels
  + currently course prescribed
* list of items borrowed/ lend

1. Draw a schematic diagram of the screen. You can do this using a pen and paper or use online android prototyping tools.



1. Now break down your design into various layouts and elements. Clearly mark what is a linear layout, what are the various elements being used in your design. For example if there is a piece of text somewhere on the screen, that part would be the “TextView”.

All the heading in the text view.

Submitted by- Shruti Singh

19D100022

Mechanical Engineering