**RESEARCH**

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

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.

In October 2003, well before the term “smartphone” was used by most of the public, and several years before Apple announced its first iPhone and its iOS, the company Android Inc was founded in Palo Alto, California. Its four founders were Rich Miner, Nick Sears, Chris White, and Andy Rubin. At the time of its public founding, Rubin was quoted as saying that Android Inc was going to develop “smarter mobile devices that are more aware of its owner’s location and preferences.” Google bought the company in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. Android is now developed by a consortium of developers known as the Open Handset Alliance, with the main contributor and commercial marketer being Google.

An Android app is a software application running on the Android platform. Because the Android platform is built for mobile devices, a typical Android app is designed for a smartphone or a tablet PC running on the Android OS. Android apps are available in the Google Play Store, in the Amazon Appstore and on various Android App-focused sites.

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

As the official integrated development environment for all Android applications, Android Studio is a good software that can be used for development.

Android Studio provides code editing, debugging, and testing tools all within an easy-to-use drag-and-drop interface. It is free to download and is supported not only by Google, but also by a large and actively engaged community of Android developers.

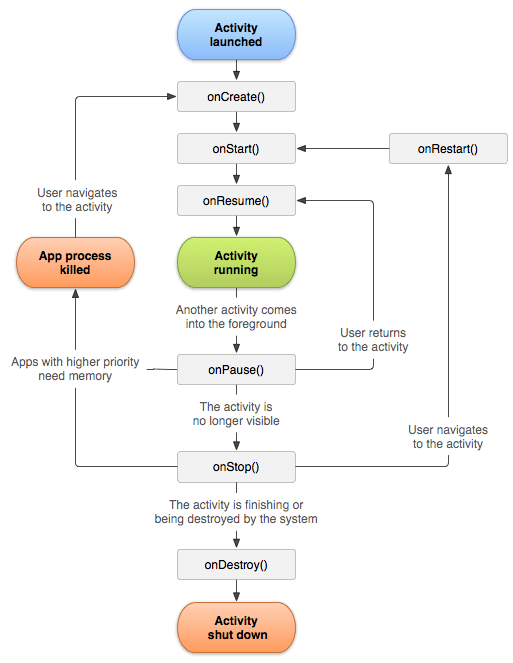
1. Which are the languages commonly used in the development of android apps? Which language does InstiApp use?

A number of languages are used in writing Android applications. The most common being Java, Python, Kotlin, C#, HTML+CSS+Javascript.

InstiApp uses python along with frameworks such as Angular and Django.

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

preferred.



The Activity lifecycle consists of 7 methods:

* onCreate() : It is called when an activity is first created. When a user opens the app then some Activity is created. You have to implement this method in every activity because, inside this method, all the necessary components of your activity will be initialized. Here the initialization of your application's UI is done.
* onStart(): This method is called when an activity becomes visible to the user. When all the initialization is done by the onCreate() method, then this method is called.
* onResume(): It is called just before the user starts interacting with the application. Most of the core functionalities of the app are implemented in this method.
* onPause():  It is called when the activity is paused i.e. it is mostly called when you press the back or home button of your Android device. It is an indication that the user is leaving the activity and starting some other activity.
* onStop(): It is called when the activity is no longer visible to the user. If you are starting a new activity, or some existing activity is entering into onResume() state, then the current activity will not be visible to the user and is stopped.
* onRestart(): It is called when the activity in the stopped state is about to start again. By doing so, the state of the activity from the time it was stopped will be restored.
* onDestroy(): It is  called when the activity is totally destroyed i.e. when you clear the application stack then onDestroy() will be called and all the states of the activity will be destroyed.

1. What are 5 different UI elements in an android app?

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, EditText, Buttons, Checkbox, Progressbar, Spinners, etc.

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

A) JAVA

1. Java is Familiar: Java is similar to C/C++ but it removes the drawbacks and complexities of C/C++ like pointers and multiple inheritances.

2. Java is an Object-Oriented programming language: Unlike C++ which is semi object-oriented, Java is a fully object-oriented programming language. It has all OOP features such as abstraction, encapsulation, inheritance and polymorphism.

3. Java supports Functional programming: Since Java SE version 8 (JDK 8), Java is updated with functional programming feature like functional interfaces and Lambda Expressions. This increases the flexibility of Java.

4. Java is Robust: It has automatic garbage collection and simple memory management model (no pointers like C/C++), plus language features like generics, try-with-resources.

5. Java is Secure: The Java platform is designed with security features built into the language and runtime system such as static type-checking at compile time and runtime checking (security manager), which let you creating applications that can’t be invaded from outside.

6. Java is High Performance: Java code is compiled into bytecode which is highly optimized by the Java compiler, so that the Java virtual machine (JVM) can execute Java applications at full speed.

7. Java is Multithreaded: The Java platform is designed with multithreading capabilities built into the language resulting in highly interactive and responsive applications.

8. Java is Platform Independence

B) PYTHON

1. Python is easy to code: Python is high level programming language. Python is very easy to learn language as compared to other language like c, c c++, java script, java etc.

2. Free and Open Source

3.Object-Oriented Language

4. GUI Programming Support: Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python.

5. High-Level Language: Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory. C++ is a lower to level language compared to python.

6.Extensible feature: We can write some python code into c or c++ language and also we can compile that code in c/c++ language.

7. Python is Portable language

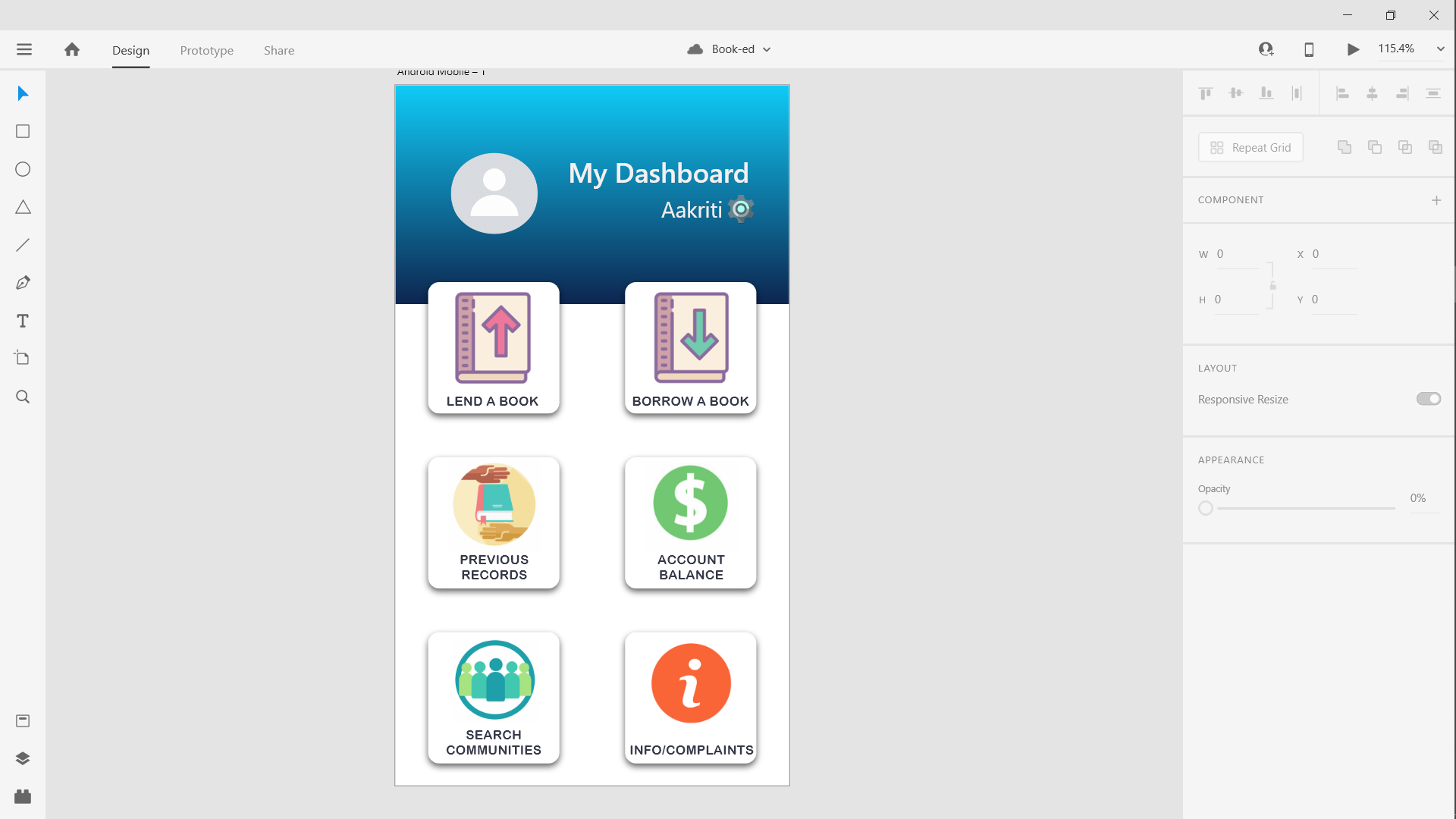
8. Python is Integrated language: Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

9. Interpreted Language

10. Large Standard Library: Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc. C++

11. Dynamically Typed Language: Variable type is decided at runtime and not prespecified.

**DASHBOARD OF BOOK-ED! CREATED IN Adobe XD**



CardView

Text Button

Linear

Layout

Image Button

Grid

Layout

Relative

Layout

ImageView

ImageView

TextView

TextView

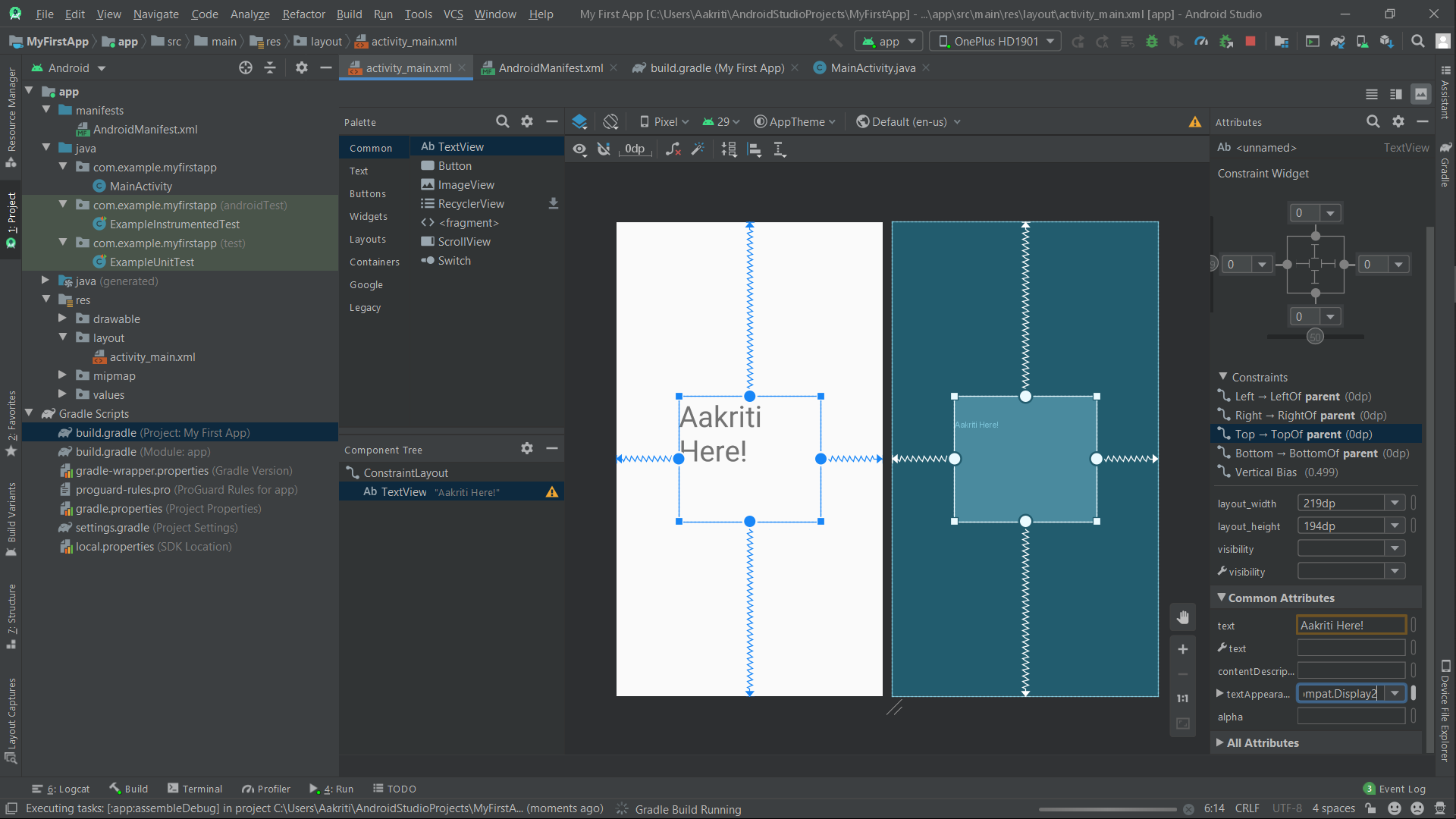
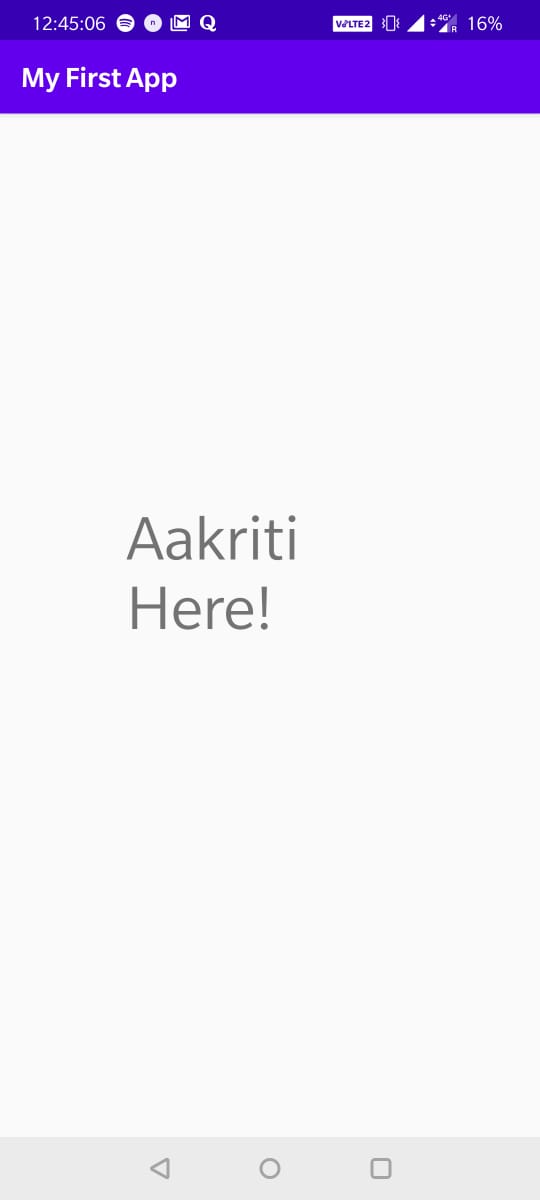
**Dashboard Features explained**

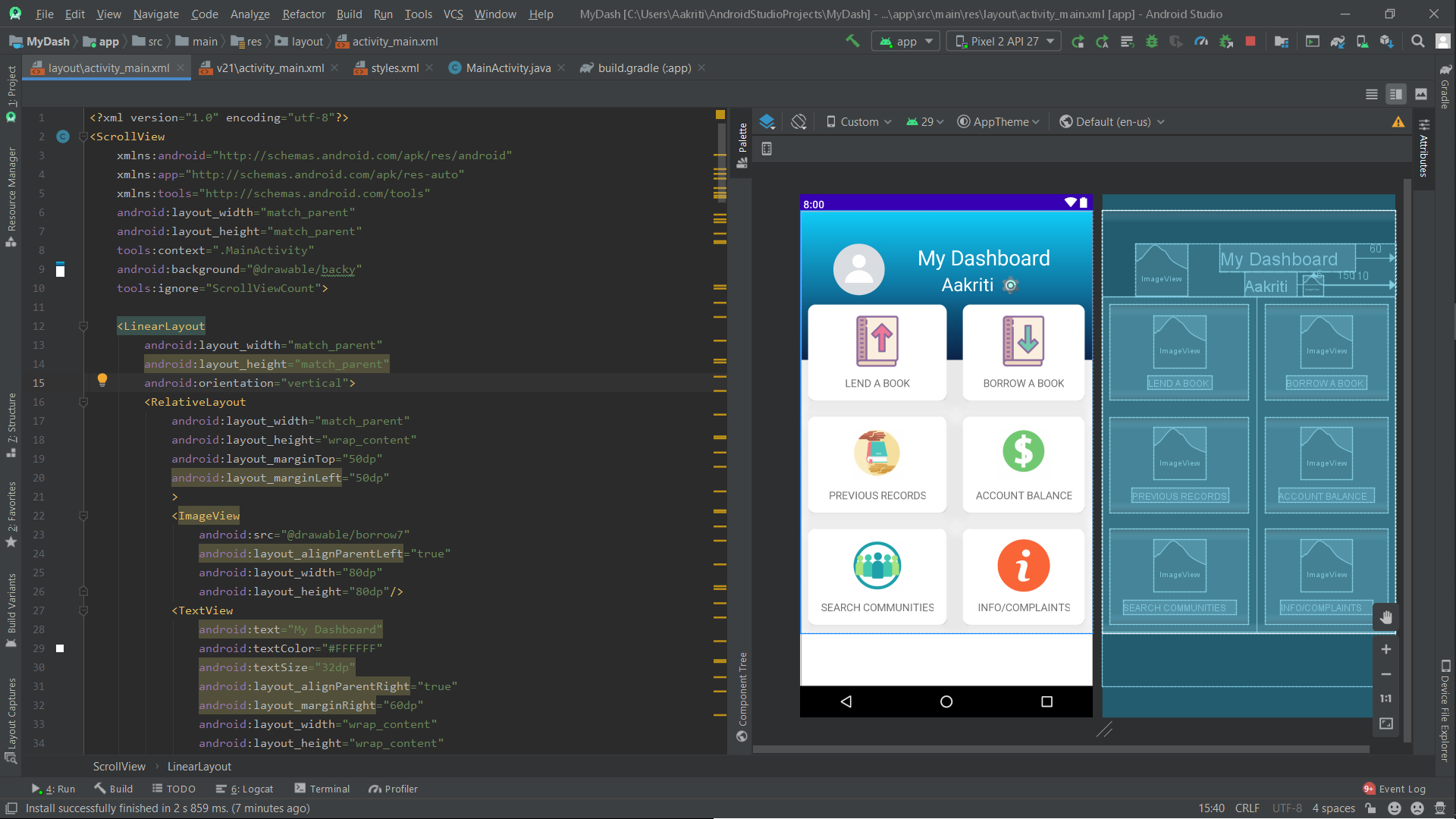
Each person has the option of joining **Communities** consisting of people who are part of same institution, organisation, club or read books of a genre or share similar interests. Communities are also formed location wise as the people would need to meet in order to exchange the book physically. The books they enlist will enlist in all communities they choose. Communities can be private or public much like Facebook public and private groups. To join a private Community you need admin’s approval. Everyone who is part of a community can scroll through the books listed in that community and borrow. Users also have option of looking up books just by the title/ author/ publication name. Matching results from all communities will be listed and you can borrow books directly from public communities or ones that you are part of. To borrow book from a private community you need to join the community by sending out a request. Books are lent out for free or for a reasonable price decided by the owner. Time period of lending is also flexible. An initial time period is listed by the lister, but books can be re-borrowed too.

Dashboard of Book-ed! Contains the following features-

1. Username or the user in a TextView
2. A Settings Button beside the username to change Account and App settings
3. A Profile Picture of the user set by the user
4. Feature to enlist a new book
5. Feature to lookup a book to borrow
6. Feature to see your past records of books listed out, books borrowed and books requested to borrow
7. Feature to check the money received through previous books lent out and payments due
8. Feature to search the available communities to lookup a specific community or scroll through the books available in a particular community
9. Feature to get information about the App or give feedback/ complaints/ suggestions

Running the starter App on Android Studio.



**DASHBOARD CREATED IN ANDROID STUDIO**

**BACKEND TASK**

1. What is the entire cycle of events that follows when you type in the url of a webpage?

The following are carried out-

* The browser extracts the domain name from the URL.
* The browser queries DNS for the IP address of the URL. Generally, the browser will have cached domains previously visited, and the operating system will have cached queries from any number of applications. If neither the browser nor the OS have a cached copy of the IP address, then a request is sent off to the system's configured DNS server. The client machine knows the IP address for the DNS server, so no lookup is necessary.
* The request sent to the DNS server is almost always smaller than the maximum packet size, and is thus sent off as a single packet. In addition to the content of the request, the packet includes the IP address it is destined for in its header. Except in the simplest of cases (network hubs), as the packet reaches each piece of network equipment between the client and server, that equipment uses a routing table to figure out what node it is connected to that is most likely to be part of the fastest route to the destination. The process of determining which path is the best choice differs between equipment and can be very complicated.
* The is either lost (in which case the request fails or is reiterated), or makes it to its destination, the DNS server.
* If that DNS server has the address for that domain, it will return it. Otherwise, it will forward the query along to DNS server it is configured to defer to. This happens recursively until the request is fulfilled or it reaches an authoritative name server and can go no further. (If the authoritative name server doesn't recognize the domain, the response indicates failure and the browser generally gives an error like "Can't find the server at www.lkliejafadh.com".) The response makes its way back to the client machine much like the request traveled to the DNS server.
* Assuming the DNS request is successful, the client machine now has an IP address that uniquely identifies a machine on the Internet. The web browser then assembles an HTTP request, which consists of a header and optional content. The header includes things like the specific path being requested from the web server, the HTTP version, any relevant browser cookies, etc. In the case in question (hitting Enter in the address bar), the content will be empty. In other cases, it may include form data like a username and password (or the content of an image file being uploaded, etc.)
* This HTTP request is sent off to the web server host as some number of packets, each of which is routed in the same was as the earlier DNS query. (The packets have sequence numbers that allow them to be reassembled in order even if they take different paths.) Once the request arrives at the webserver, it generates a response (this may be a static page, served as-is, or a more dynamic response, generated in any number of ways.) The web server software sends the generated page back to the client.
* Assuming the response HTML and not an image or data file, then the browser parses the HTML to render the page. Part of this parsing and rendering process may be the discovery that the web page includes images or other embedded content that is not part of the HTML document. The browser will then send off further requests (either to the original web server or different ones, as appropriate) to fetch the embedded content, which will then be rendered into the document as well.

1. How are urls mapped to website hosted on a server in Bangalore?

General-purpose webservers like Apache start out with files and directories, so they split up the URL into a hierarchical path description, try to find a file at the given location, and serve it if it exists. (This gets more complex with modules and filetypes; some filetypes imply processing the file as a script and returning the script output rather than just piping out the file contents, and so on).

The webserver kept in Bangalore extracts the relevant part from the URL, converts it to an absolute disk file system path based on the path to the web document root, locates the file on disk and streams it. Most web-servers have the notion of "virtual directories" (which exist as part of the URL) which are mapped to some sort of physical path and/or application. Exactly how the mapping occurs is very web-server specific.

1. Read up in brief about TCP and UDP, that enable your request to reach a particular server, hopping from one router to another.

The Internet works by using a protocol called TCP/IP, or Transmission Control Protocol/Internet Protocol. TCP/IP is the underlying communication language of the Internet. In base terms, TCP/IP allows one computer to talk to another computer via the Internet through compiling packets of data and sending them to right location. In fact, each packet could take a different route to the other computer, if the first route is unavailable or congested. In addition, TCP/IP divides the different communications tasks into layers. Each layer has a different function. Data goes through four individual layers before it is received on the other end. TCP/IP then goes through these layers in reverse order to reassemble the data and to present it to the recipient.

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of Internet Protocol suite, referred as UDP/IP suite. Unlike TCP, it is unreliable and connectionless protocol. So, there is no need to establish connection prior to data transfer. Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of Internet services; provides assured delivery, reliability and much more but all these services cost us with additional overhead and latency. Here, UDP comes into picture. For the real-time services like computer gaming, voice or video communication, live conferences; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also saves bandwidth. User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

1. What does setting up a server even mean?

Setting up a server On the hardware side means, physically installing a computer that stores web server software and a website's component files (e.g. HTML documents, images, CSS stylesheets, and JavaScript files). It is connected to the Internet and supports physical data interchange with other devices connected to the web.

On the software side setting up a server means, installing several programs that control how web users access hosted files, at minimum an HTTP server. An HTTP server is a piece of software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages). It can be accessed through the domain names (like mozilla.org) of websites it stores, and delivers their content to the end-user's device.

1. Get an overall idea of how a server resolves a request? [ Google Apache and NginX, and

see what role they play].

Apache and Nginx are the two most common open source web servers in the world. Together, they are responsible for serving over 50% of traffic on the internet. Both solutions are capable of handling diverse workloads and working with other software to provide a complete web stack.

Mapping a domain name to an IP Address is known as Name-Address Resolution. The Domain Name Server (DNS) Resolver performs this operation by consulting name servers.

In order to find a particular DNS the requesting host place it’s query to the Local DNS Server with a mapping request. If it has the information, the resolver is satisfied else the resolver is referred to other servers or other servers are asked to provide the information. After the resolver, gets the response, it checks whether the response is correct or not. If the response is correct, the response is passed to the process that requested it, else the name query fails. A resolution can be of two types – iterative and recursive.