

**GLA UNIVERSITY, MATHURA
MINI PROJECT
(2020-2021)**

BOOK FINDER APP

MID-TERM REPORT



**GLA University, Mathura
Institute of Engineering and Technology**

Submitted By:-

- 1.Ashita Goyal (181500144)
- 2.Bharti Gautam(181500192)
- 3.Janvi Pangoriya(181500292)
- 4.Nidhi Gupta(181500422)
5. Naina Agarwal(181500408)

**Under the Supervision
Of**

Prof. Mandeep Singh
Technical Trainer

Department of Computer Engineering and Applications

Contents

Abstract

- 1. Introduction**
- 2. Problem Definition**
- 3. Objectives**
- 4. Implementation Details**
- 5. Progress till Date and the remaining work**
- 6. Some Screenshots**

References

Abstract

In this project, we are creating an android application, basically a Book Finder App which we have named Bookopedia. This application will provide us a platform to access the books we want to read at the ease of our fingertips. All the users will be having their separate accounts on this app which will be connected to their email id. Any book that the user wishes to read, will be entered by him in the search box which works on the basis of queries input. The query may be an author name, the book name or the subject to which the book is related and will also work on specific keywords input. Apart from searching the books online, the user can save the book he/she likes in the favourites bar. The app is suitable in the present scenario as the world is being digitalized then why not the education system.. On the profile of the user, one can easily view the books he/she has read. The app will be completely efficient and transparent to the reviews of the people on the book and its price. To get more details about the book one can click on the book and get further grave details. This app will be using The Google Book API for providing all the books. Further the FAQ section provides all the necessary details that the user may need about the app. The app also has a complete User Interface attached to the firebase a perfect login system with email id and password and a forget password too.

Android App ecosystem is diverse and is changing people's life all over the world. Android users are expected to increase because of the advance changes of the operating system and the way it deals with issues and compatibility with other mobile devices. Furthermore designing solutions for the problems that we may face in future is essential. Like this application definitely stands the need of students at any time at their fingertips without any barrier of place.

Introduction

- **General Introduction to the Topic**

Bookopedia, an android application is a user friendly app, with a well-developed User Interface, a login module using the email address and the password and a sign up system for the new users and provides the user the ability to search for any book one wants to read.

Besides the user Interface of the Login system, we have also created a navigation bar to provide various other features that the app provide, It has various fragments the profile page, the favourites coloumn, About us section, FAQ section, and an option to Sign Out from the account recently logged in. We have abstracted the android application using the Google Book API which supports us enabling the various books to the users they are searching for.

We felt the need of developing this app looking at the current scenario where it is impossible to learn or get a book from the library and so the user can access the book from the app store the book whichever one likes on the cloud storage available for the profile.

For testing our application, at last we will connect the app with the Google Book API and will allow a user to create a new account and access all the features of the app. This is all our app does and will be the outcome of our project on completion.

Key Features of the app – Bookopedia

1. A “Welcome Page (Splash Page)” which displays the logo and the name of the app.
2. A “Login Page” which asks the users email id and password.
3. A “Registration Page” where our users can create an account for the app.
4. A “Search page” where user can enter the query.

5. A “Book list page” which will contain all the recommended books.
6. A “Book description page “where the details of the book the user liked will be displayed.
7. A forgot password page which will enable the user to reset the password.
8. A “Profile Page” where the user information will be displayed.
9. A “Favourite Page” where all the books marked as favourite by the user will be displayed.
10. A “FAQ Page” that will display frequently asked questions.

How the Bookopedia App works?

- As soon as the user enters there would be a splash screen displaying the app name and the copyrights reserved.
- Then there would be a login page for the user to enter the email address and the password as set if the user has already signed up. For the new users there will be a sign-up page.
- After logging in, there will be the search page having the search bar, an opportunity for user to enter any keyword related to the book.
- As the app will be connected with the Google Book API the resulted books found will be displayed on the page itself.
- According to interest and the choice of the user ,the user can click on the book and know details about the book(the price, the author name, the reviews of the book, about what the book is all about and even may be the pdf copies of the book if we can)
- The user will have the choice to read the book then and there on the spot or can add it to the favourite section.
- The favourite section will store the book till the user wants .
- For any help or details about how the app functions one can visit the FAQ section .
- After the work is done user can sign out from the account.

Technologies Used

- **Kotlin:**
While developing an android app it is very important to code the logic for things to happen(Backend framework) .It is important for the app to understand what to perform on what commands. To make this happen

we use programming languages .For android app development two languages have been used since times – Java and Kotlin.

In our project we have been using Kotlin components because of the following:-

>> Java is old, verbose, and contains a lot of boiler plate code and that is difficult to cope up with. So Kotlin comes as a best alternative which allows a way lesser coding and has lesser boiler plate code.

>> One of the main reasons for Kotlin to grow is Interoperability.it can co-exist with its elder sibling java on the same project.

>> No cost adoptions as Kotlin is an open source language, statically typed and based on Java Virtual Machine.

>>Kotlin is safer as it doesn't allow you to have Null Pointer Exceptions. The compiler is designed to auto-cast mistakes.

- **XML**

Extensible Markup Language it is a meta-language which allows users to define their own customized markup language, especially in order to display documents, layouts on internet. It is the language that contains tag to store up the information and is used to present the data on the screen. XML is used for the User Interface of our app.

We will use different XML components and Kotlin components that would combine us to give the Android Application.

- **Hardware and Software Requirements**

- Hardware Requirements**

- Processor :intel i5
 - Operating System :Windows 10
 - RAM : 8 GB
 - Hard disk : 256GB

- Software Requirements**

- Android Studio
 - Database: Firebase
 - User Interface Design: Android Application

Problem Statement

The unpredictable period we face, such as lockdown, it is impossible to arrange the hard copies of the book so application of virtual education can reduce the barrier to get knowledge at any place in a cost effective, productive way. We have realized the importance of virtual learning and how important it is for us to have our resources online. And along with this we need to have a place to keep the resources for areas of our interest so we thought of developing this app and hence implementing it. To solve this problem and to contribute to the virtual application we introduce you with the concept of Bookopedia.

For students who are interested in learning online can use this application and keep all the books they want to learn from at one place (in favourites section) and can create their own personal E-library. Imparting Education complying with the new Educational Reforms and hence saving the paper that is used in making books. Even individual book stores can have this system of book apps promoting their brand name as Digital Marketing and can gain number of customers.

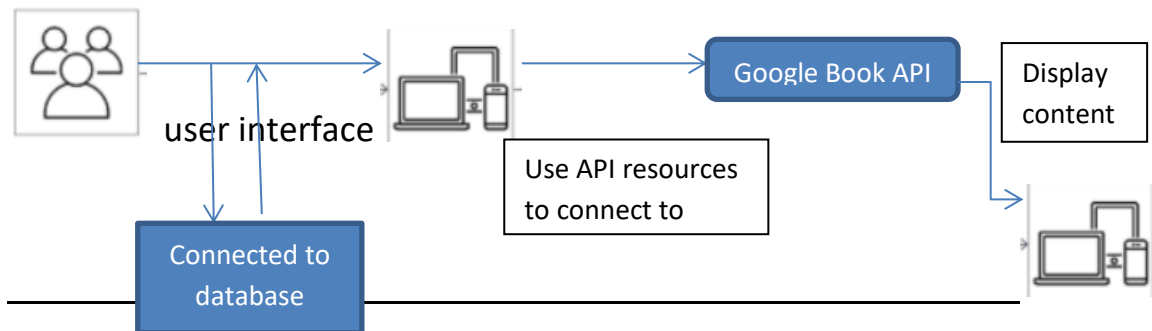
Objectives

The objective of the app Bookopedia is to be a source of Virtual education allowing its users to search the book of their choice according to the criteria set by them in the search page ,read the book ,add it to the favourite section, see the reviews of the book and decide which can be better source for their knowledge and store them at the storage of their profile's cloud. The app will be transparent and efficient in searching all the books it could as available on the Google Book API. Furthermore new books will be added automatically as it will be updated on the API.

This app will be free of cost so all the students interested to learn can easily learn from it. Moreover this kind of application can be used in areas/schools where guardians /parents cannot afford to buy books. This would be an excellent effort to provide education without any boundaries to all.

Implementation

The Android App created hold on to the following steps:



Part 1: To create the User Interface, we have used the XML language and created layouts for each page-the splash page , the login page, sign up page, forget password page and connected them all with the database .

Part 2: After creating all these layouts we have made the navigation bar to access all the features of the app efficiently and we have created fragments in that navigation bar :the dashboard, the profile , the favourite bar , the FAQ section , the About Us section and the sign out section.

Part 3: After creating the fragments we have done the respective parts of each fragments accordingly like putting up FAQ questions, showing the user profile , enabling the user to edit their profile, seeing the books in the favourite bar.

Part 4: the last step which includes connecting the app to the Google Book API so as to provide end to all tasks and see the working of the app properly to see its usability and credibility.

Whole of the user interface is created using the XML in which all the pages are connected to each other which in turn is connected to the firebase. The backend part coded using kotlin calls up the Google Book API each time the user enters the search query and brings up all the related books it get in API to display on the search page and on further clicking on to it display more details as loaded by the browser.

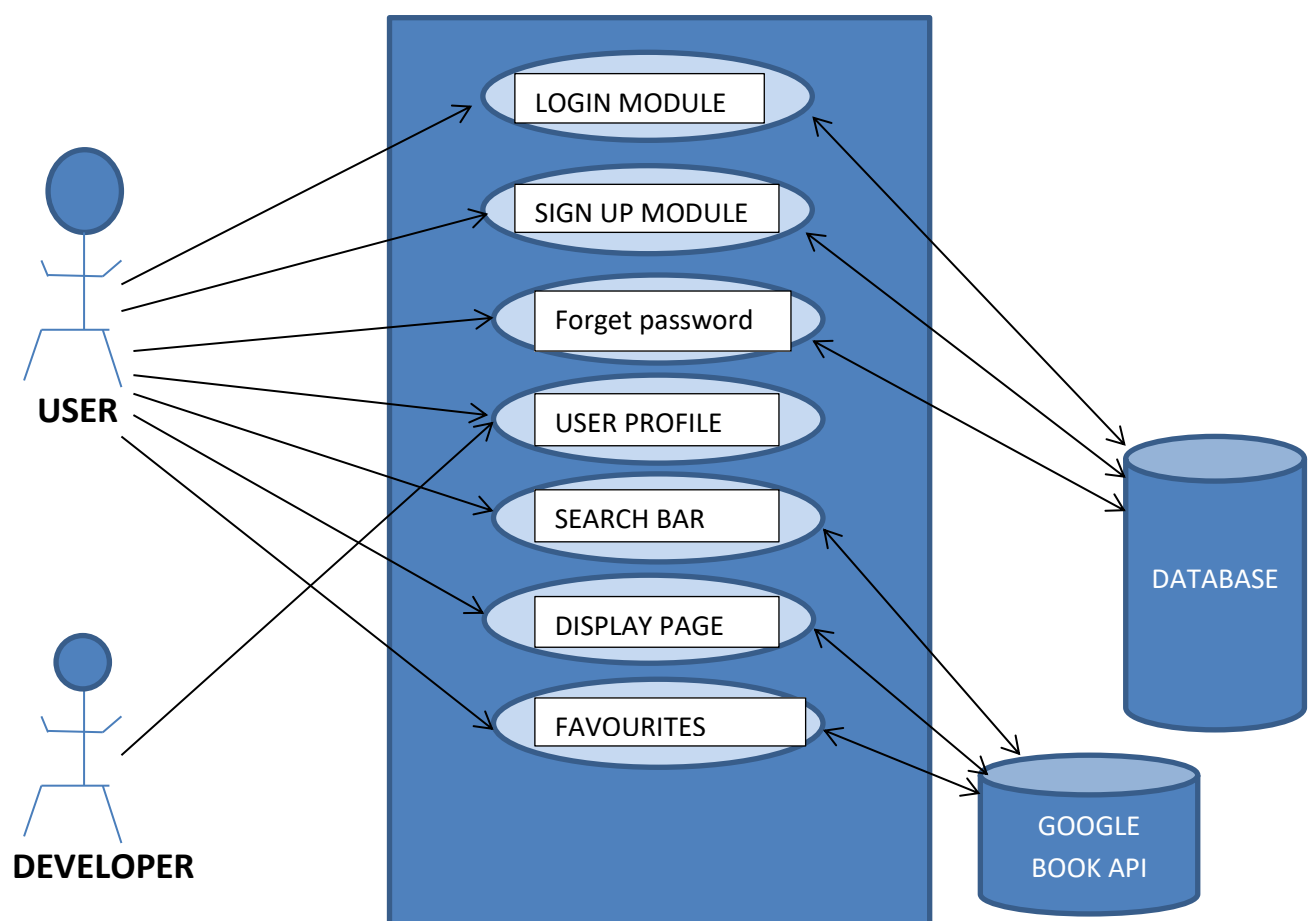
It is also worth noting that since we are using the Google book API for searching the book user is asking results for, we do not need to add the book in

API or update it when a new book arises but it automatically gets updated. All the login data is being retrieved from the database connected.

The app can track the fake login users and tell them if they are trying to login by using forget password as the forget password module before sending the mail to the user to recover the password checks whether the mail id is present in database or not.

The app is completely non-profitable and solely for use of free virtual education.

USE CASE DIAGRAM FOR THE MODEL



Progress

Part 1: (done)

Creating all the layouts using XML and connecting to database is **done**. This includes:-

- Creating login page
- Creating Sign up page
- Creating forget password page
- Creating Flash page with copyrights
- Registering login using Firebase Auth.

Part 2: (done)

Connecting all the activities using Kotlin and creating a navigation bar to access all those Activities efficiently **is done**. This includes:-

- Connecting activities using Kotlin
- Navigation Bar created
- Fragments added (dashboard, profile, FAQ, About us, Favourite)

Part 3: (done)

Creating the search page with sample books and adding the necessary details. Moreover content to the fragments in the navigation bar. Basically includes:-

- Search page created
- Sample book added
- User profile created
- Question answer added to FAQ
- Favourites bar work done
- About us part is complete

Part 4: (YET TO BE DONE)

This part includes :-

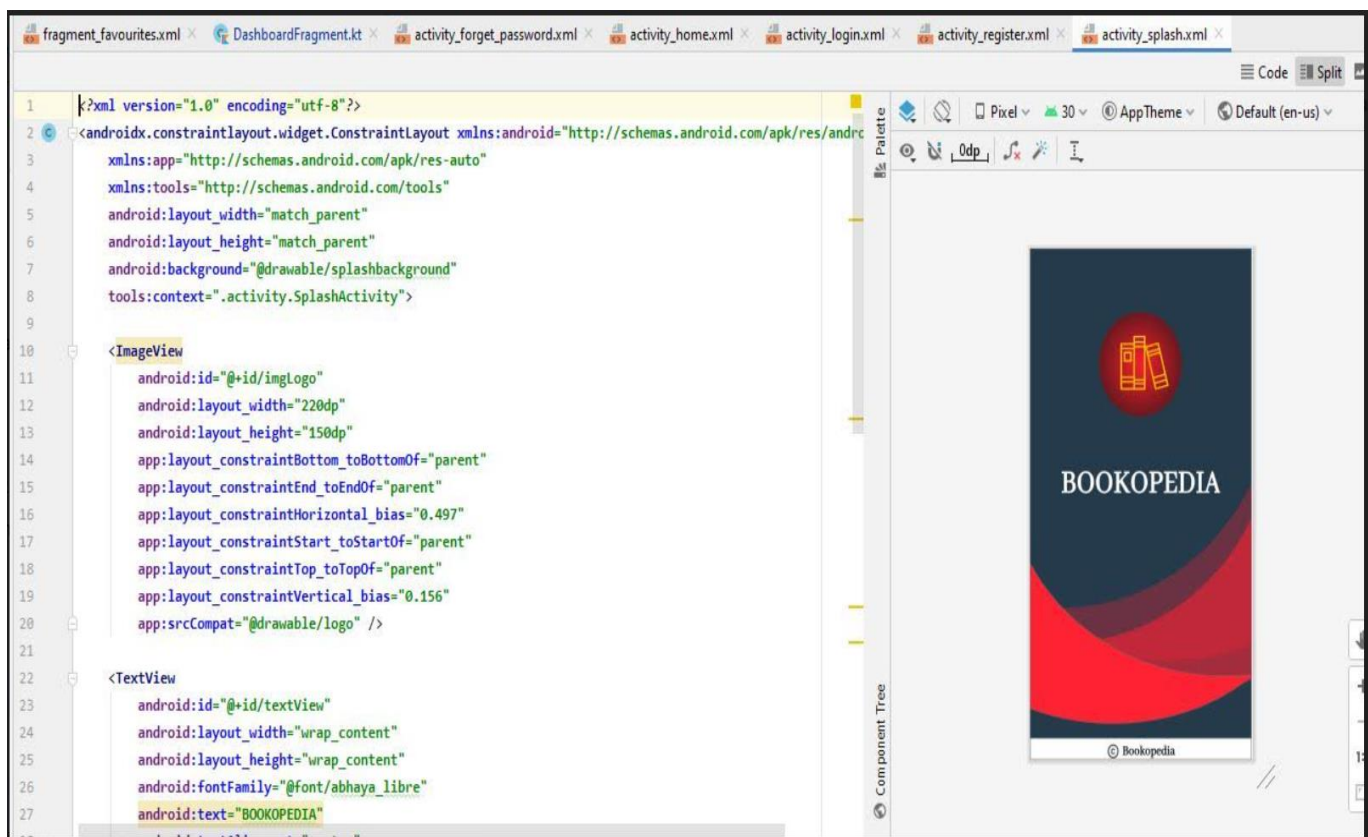
- Connecting the Google API to the app
- Displaying the books as received on queries from API
- Enabling to provide the details of the book to the user

SCREENSHOTS OF THE WORK COMPLETED

PART 1:

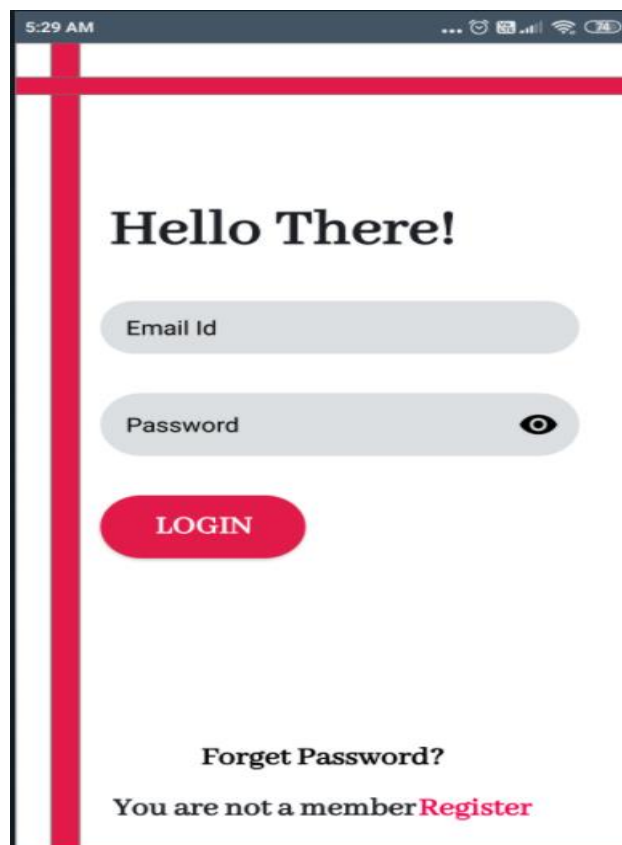
SPLASH PAGE:

LAYOUT CODE



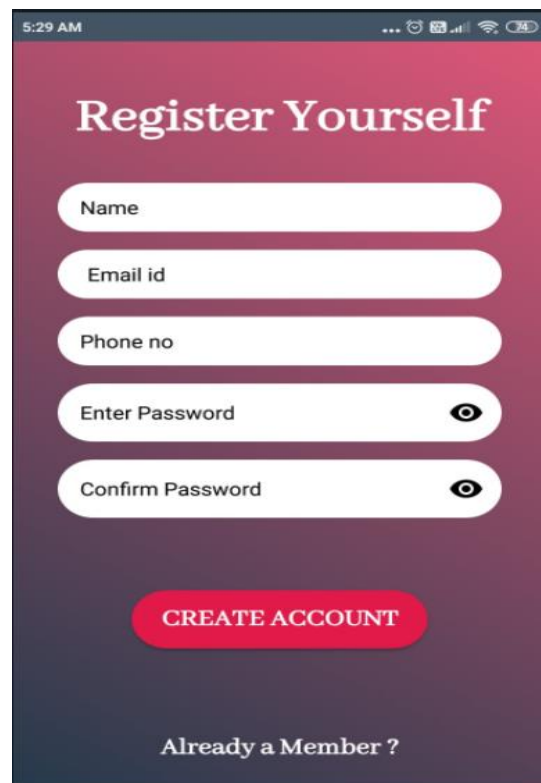
2. LOGIN PAGE (LAYOUT)

```
fragment_favourites.xml x DashboardFragment.kt x activity_forget_password.xml x activity_home.xml x activity_login.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="@drawable/loginbackground"
8     tools:context=".activity.LoginActivity">
9
10    <TextView
11        android:id="@+id/main"
12        android:layout_width="wrap_content"
13        android:layout_height="wrap_content"
14        android:layout_marginLeft="55dp"
15        android:layout_marginTop="120dp"
16        android:fontFamily="@font/abhaya_libre"
17        android:text="Hello There!"
18        android:textColor="#1B1E23"
19        android:textSize="40sp"
20        android:textStyle="bold" />
21
22
23    <EditText
24        android:id="@+id/etEmail"
25        android:layout_width="match_parent"
26        android:layout_height="wrap_content"
27        android:layout_below="@id/main"
28        android:layout_marginLeft="55dp"
29        android:layout_marginTop="20dp"
30        android:background="@drawable/whitebackground"
31        android:padding="10dp" />
32
33    <EditText
34        android:id="@+id/etPassword"
35        android:layout_width="match_parent"
36        android:layout_height="wrap_content"
37        android:layout_below="@id/etEmail"
38        android:layout_marginLeft="55dp"
39        android:layout_marginTop="10dp"
40        android:background="@drawable/whitebackground"
41        android:padding="10dp"
42        android:password="true" />
43
44    <Button
45        android:id="@+id/login"
46        android:layout_width="150dp"
47        android:layout_height="40dp"
48        android:layout_marginLeft="55dp"
49        android:layout_marginTop="20dp"
50        android:background="@drawable/loginbutton"
51        android:text="LOGIN" />
52
53    <TextView
54        android:id="@+id/forget_password"
55        android:layout_width="150dp"
56        android:layout_height="30dp"
57        android:layout_marginLeft="55dp"
58        android:layout_marginTop="20dp"
59        android:background="@drawable/whitebackground"
60        android:padding="5dp"
61        android:text="Forget Password?" />
62
63    <TextView
64        android:id="@+id/register"
65        android:layout_width="150dp"
66        android:layout_height="30dp"
67        android:layout_marginLeft="55dp"
68        android:layout_marginTop="10dp"
69        android:background="@drawable/whitebackground"
70        android:padding="5dp"
71        android:text="You are not a member Register" />
72
73 </RelativeLayout>
```



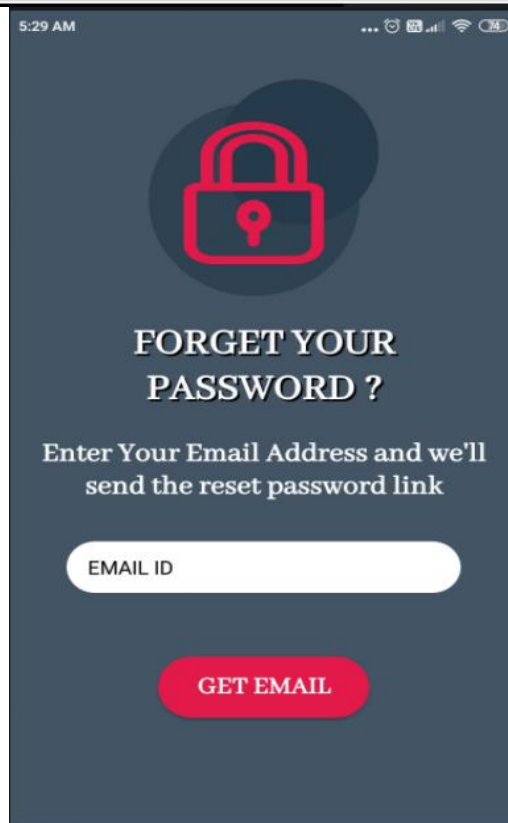
SIGN UP PAGE FOR NEW USERS (LAYOUT)

```
fragment_favourites.xml x DashboardFragment.kt x activity_forget_password.xml x activity_home.xml x activity_login.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/register_layout"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:background="@drawable/registerbackground"
9     tools:context=".activity.RegisterActivity">
10
11     <TextView
12         android:id="@+id/textregister"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:layout_centerHorizontal="true"
16         android:layout_marginTop="40dp"
17         android:fontFamily="@font/abhaya_libre"
18         android:text="Register Yourself"
19         android:textAlignment="center"
20         android:textColor="#ffffff"
21         android:textSize="40sp"
22         android:textStyle="bold" />
23
24     <EditText
25         android:id="@+id/edtname"
26         android:layout_width="match_parent"
27         android:layout_height="wrap_content"
```



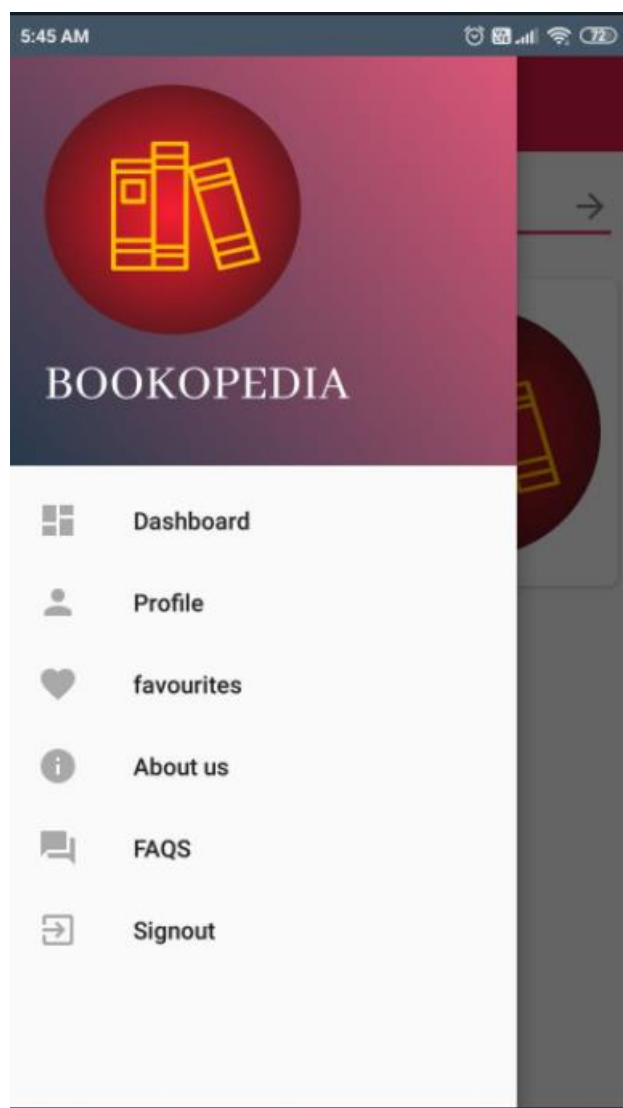
FORGOT PASSWORD (LAYOUT)

```
fragment_favourites.xml x DashboardFragment.kt x activity_forget_password.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/forgetpassword"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:background="@color/grey"
8   tools:context=".activity.ForgetPasswordActivity">
9
10  <ImageView
11    android:id="@+id/imgforget"
12    android:layout_width="180dp"
13    android:layout_height="170dp"
14    android:layout_centerHorizontal="true"
15    android:layout_marginTop="30dp"
16    android:src="@drawable/locklogo" />
17
18  <TextView
19    android:id="@+id/edtforgettext"
20    android:layout_width="match_parent"
21    android:layout_height="wrap_content"
22    android:layout_below="@id/imgforget"
23    android:layout_centerHorizontal="true"
24    android:layout_marginLeft="30dp"
25    android:layout_marginTop="20dp"
26    android:layout_marginRight="30dp"
27    android:fontFamily="@font/abhaya_libre"
```



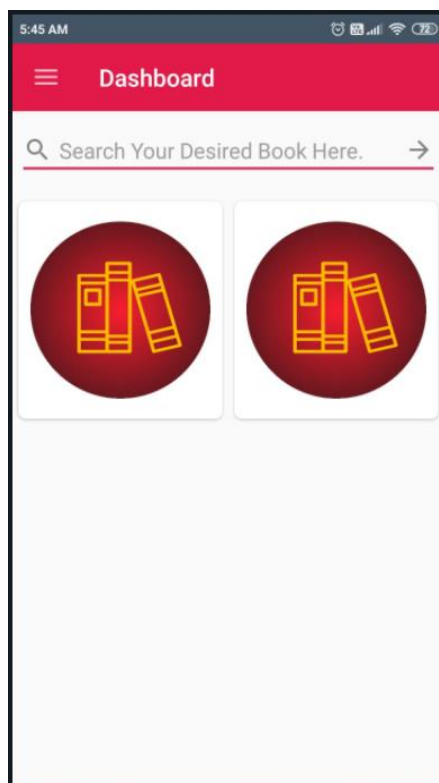
PART 2

NAVIGATION BAR



SEARCH PAGE WITH THE SEARCH BAR (LAYOUT)

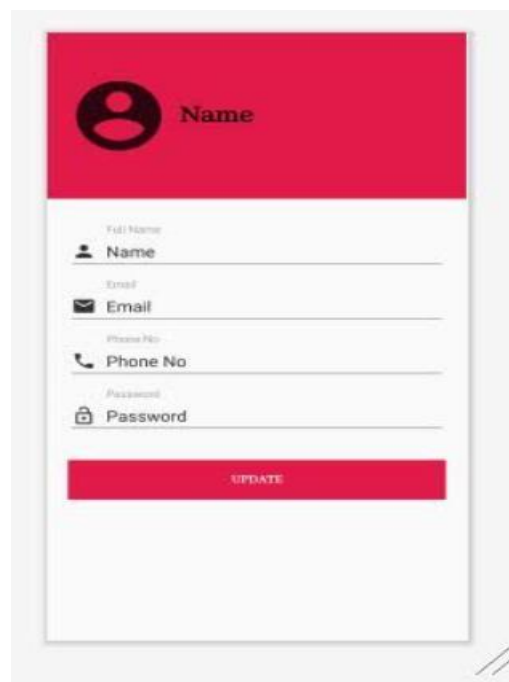
```
fragment_favourites.xml x fragment_dashboard.xml x fragment_faqs.xml x
6      android:layout_height="match_parent"
7      tools:context=".fragment.DashboardFragment">
8
9      <androidx.constraintlayout.widget.ConstraintLayout
10         android:layout_width="match_parent"
11         android:layout_height="match_parent">
12
13         <LinearLayout
14             android:layout_width="match_parent"
15             android:layout_height="match_parent"
16             android:orientation="vertical">
17
18             <androidx.constraintlayout.widget.ConstraintLayout
19                 android:layout_width="match_parent"
20                 android:layout_height="match_parent">
21
22                 <EditText
23                     android:id="@+id/edtSearch"
24                     android:layout_width="match_parent"
25                     android:layout_height="wrap_content"
26                     android:layout_marginTop="10dp"
27                     android:layout_marginStart="5dp"
28                     android:layout_marginEnd="5dp"
29                     android:textSize="18sp"
30                     android:drawableStart="@drawable/ic_action_search"
31                     android:drawableEnd="@drawable/ic_action_search_enter"
32                     android:drawablePadding="6dp"
```



PART 3: COMPLETING THE FRAGMENTS

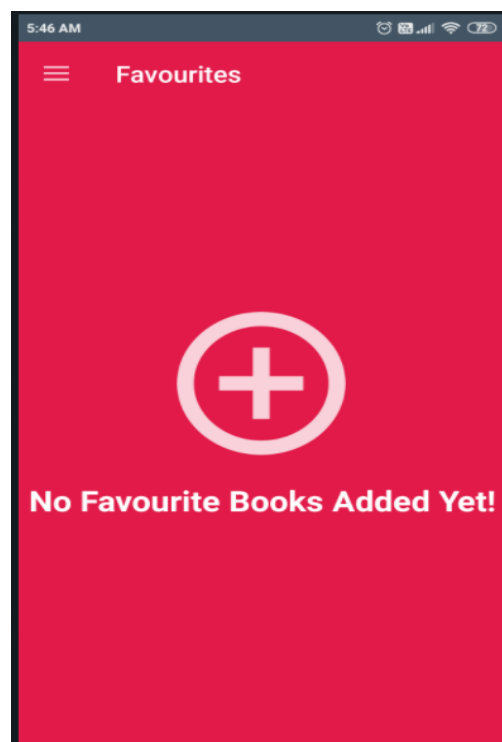
- PROFILE (LAYOUT)

```
fragment_favourites.xml x fragment_dashboard.xml x fragment_faqs.xml x fragment_profile.xml x
1 |<?xml version="1.0" encoding="utf-8"?>
2 |<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3 |    xmlns:app="http://schemas.android.com/apk/res-auto"
4 |    xmlns:tools="http://schemas.android.com/tools"
5 |    android:layout_width="match_parent"
6 |    android:layout_height="match_parent"
7 |
8 |    tools:context=".fragment.ProfileFragment">
9 |
10 |    <LinearLayout
11 |        android:layout_width="match_parent"
12 |        android:layout_height="match_parent"
13 |        android:orientation="vertical">
14 |
15 |        <RelativeLayout
16 |            android:layout_width="match_parent"
17 |            android:layout_height="200dp"
18 |            android:background="@color/pink"
19 |            android:padding="20dp">
20 |
21 |            <ImageView
22 |                android:id="@+id/profile_image"
23 |                android:layout_width="100dp"
24 |                android:layout_height="100dp"
25 |                android:layout_centerVertical="true"
26 |                android:src="@drawable/ic_profileimage" />
27 |
```

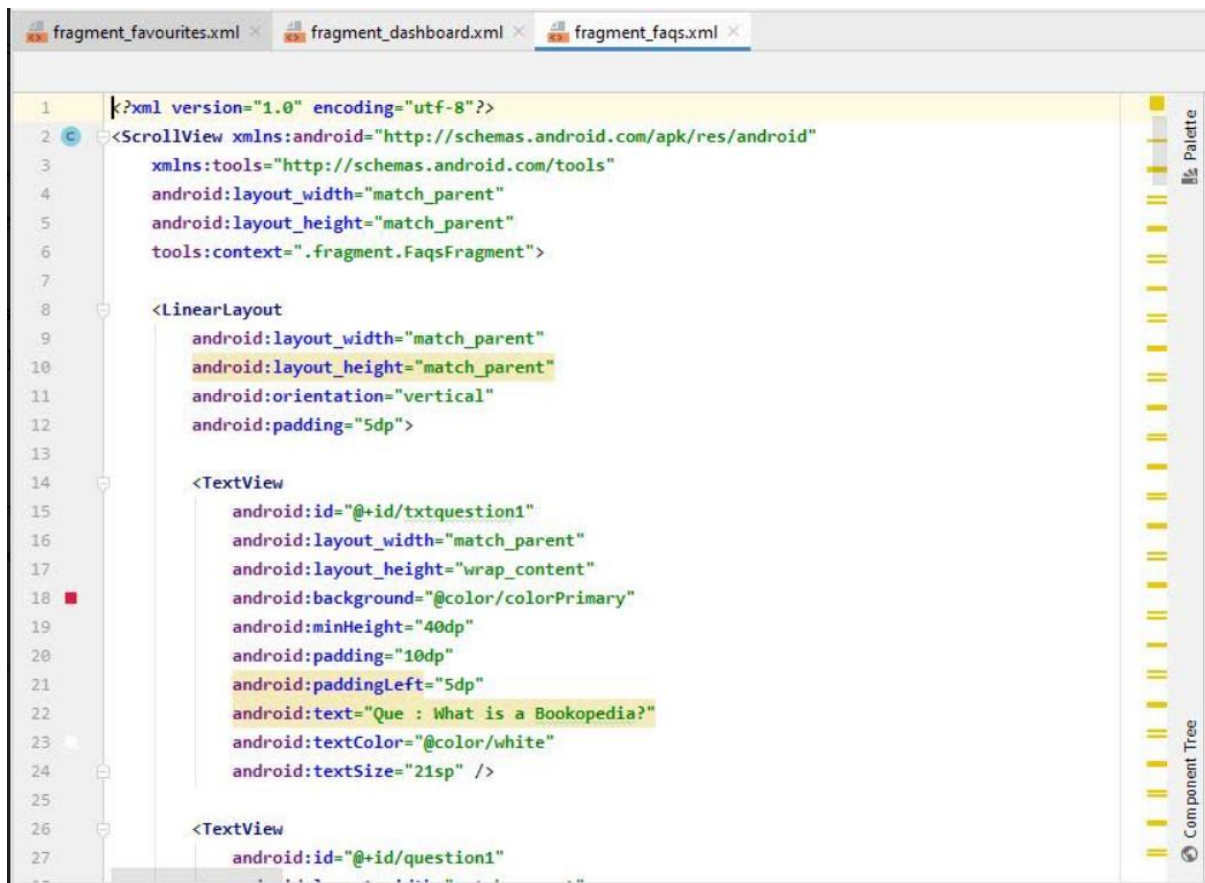


2. FAVOURITES FRAGMENT (LAYOUT)

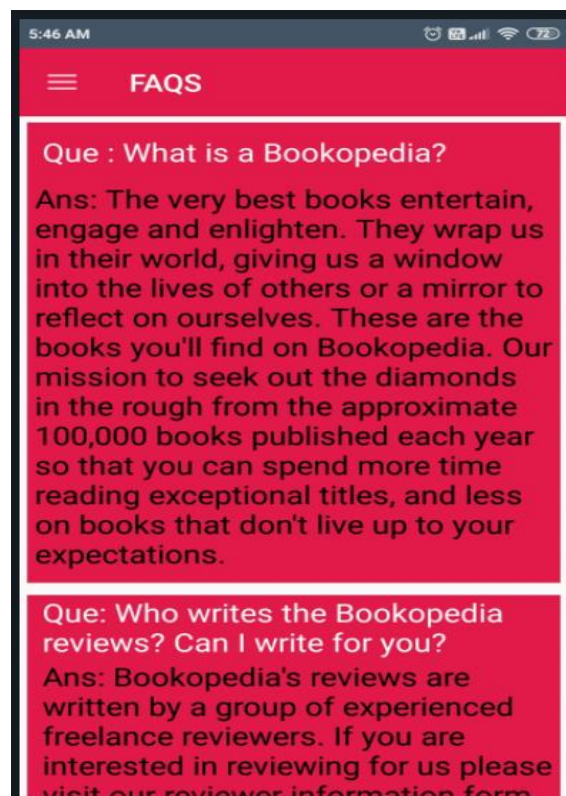
```
fragment_favourites.xml × fragment_dashboard.xml × fragment_faqs.xml ×
1 |<?xml version="1.0" encoding="utf-8"?>
2 |<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
3 |    xmlns:app="http://schemas.android.com/apk/res-auto"
4 |    xmlns:tools="http://schemas.android.com/tools"
5 |    android:layout_width="match_parent"
6 |    android:layout_height="match_parent"
7 |    tools:context=".fragment.FavouritesFragment">
8 |
9 |    <RelativeLayout
10 |        android:id="@+id/r1NoFavContent"
11 |        android:layout_width="match_parent"
12 |        android:layout_height="match_parent"
13 |        android:background="@color/pink"
14 |        android:visibility="gone">
15 |
16 |        <androidx.constraintlayout.widget.ConstraintLayout
17 |            android:layout_width="match_parent"
18 |            android:layout_height="match_parent">
19 |
20 |            <ImageView
21 |                android:id="@+id/imgEmptyFavourites"
22 |                android:layout_width="150dp"
23 |                android:layout_height="150dp"
24 |                android:layout_centerHorizontal="true"
25 |                android:layout_marginStart="261dp"
26 |                android:layout_marginTop="200dp"
27 |                android:layout_marginEnd="261dp"
28 |                android:layout_marginBottom="200dp"
29 |                android:src="@drawable/ic_add"
30 |            />
31 |
32 |            <TextView
33 |                android:id="@+id/tvEmptyFavourites"
34 |                android:layout_width="wrap_content"
35 |                android:layout_height="wrap_content"
36 |                android:layout_centerHorizontal="true"
37 |                android:layout_marginTop="20dp"
38 |                android:text="No Favourite Books Added Yet!"
39 |            />
40 |        />
41 |    />
42 |</FrameLayout>
```

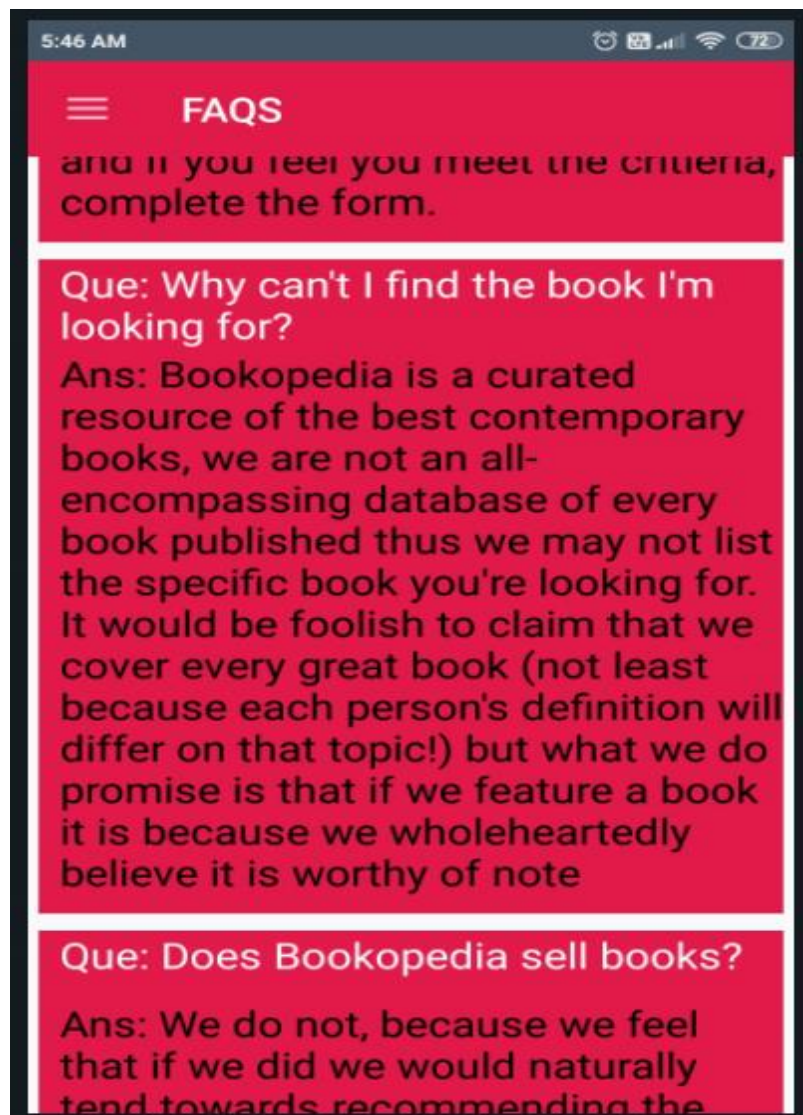


FAQ SECTION (LAYOUT)



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     tools:context=".fragment.FaqsFragment">
7
8     <LinearLayout
9         android:layout_width="match_parent"
10        android:layout_height="match_parent"
11        android:orientation="vertical"
12        android:padding="5dp">
13
14        <TextView
15            android:id="@+id/txtquestion1"
16            android:layout_width="match_parent"
17            android:layout_height="wrap_content"
18            android:background="@color/colorPrimary"
19            android:minHeight="40dp"
20            android:padding="10dp"
21            android:paddingLeft="5dp"
22            android:text="Que : What is a Bookopedia?"
23            android:textColor="@color/white"
24            android:textSize="21sp" />
25
26        <TextView
27            android:id="@+id/question1"
```





About us section

5:46 AM



About Us

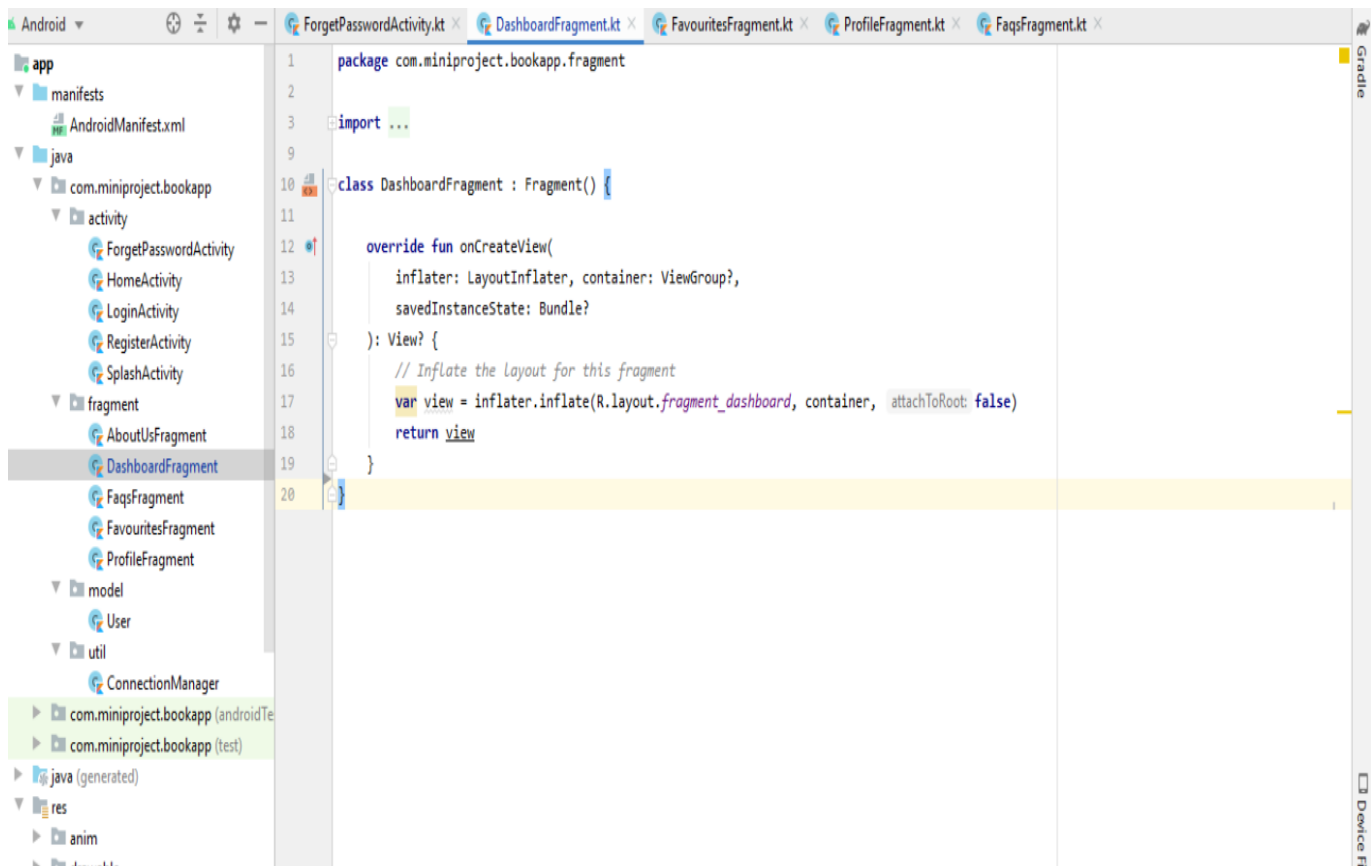
BOOKOPEDIA

About App

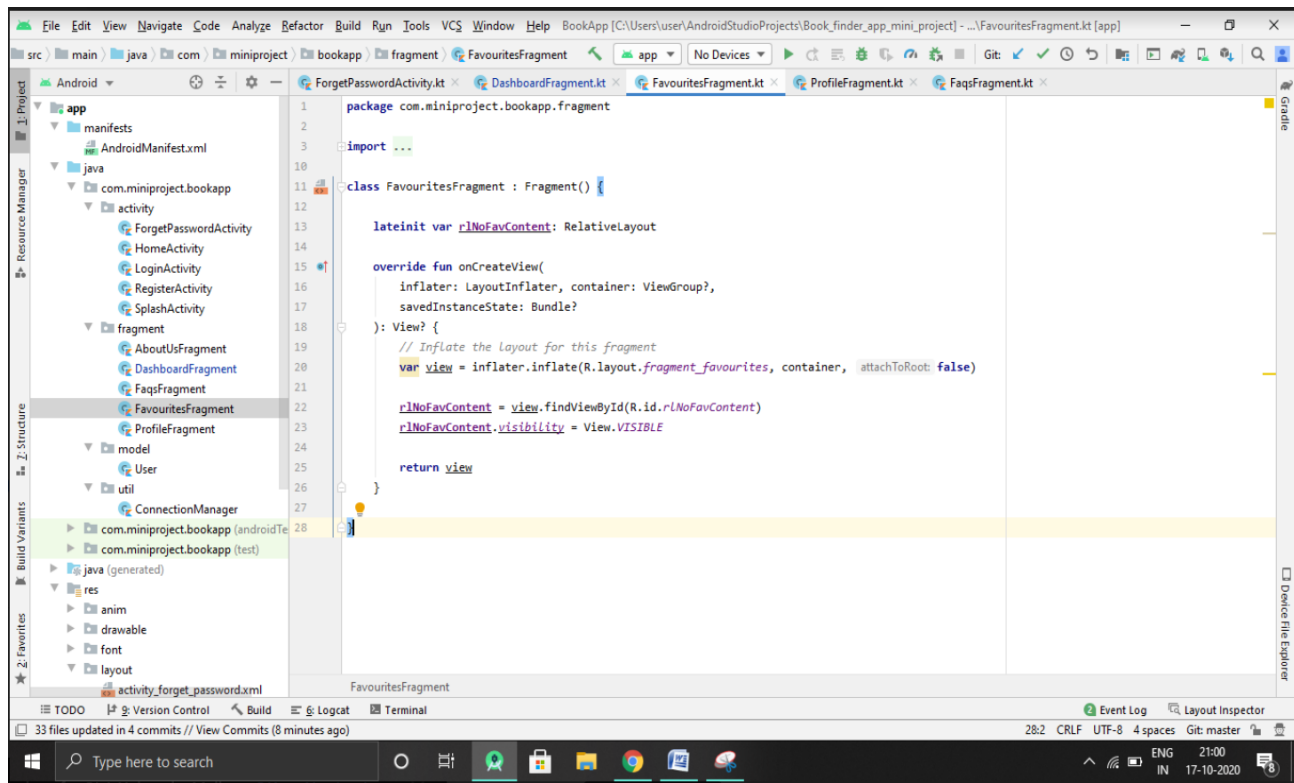
It is a multidisciplinary platform to access the books of different genres. This app is easy to use and easy to access. Our app will give readers new ways to show their love for books, review them online, keep track of what they read and of course, discover new books to read.

KOTLIN CODE

- Dashboard code



- Fragments code
Favourite fragment



LOGIN MODULE KOTLIN CODES

SPLASH PAGE

app > src > main > java > com > miniproject > bookapp > activity > SplashActivity

app No Device

HomeActivity.kt ForgetPasswordActivity.kt LoginActivity.kt RegisterActivity.kt SplashActivity.kt

1: Project

Resource Manager

Z: Structure

Build Variants

2: Favorites

```
1 package com.miniproject.bookapp.activity
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.os.Handler
7 import android.view.animation.Animation
8 import android.view.animation.AnimationUtils
9 import android.widget.ImageView
10 import android.widget.TextView
11 import com.miniproject.bookapp.R
12
13 class SplashActivity : AppCompatActivity() {
14
15     lateinit var logo: Animation
16     lateinit var name: Animation
17     lateinit var imgLogo: ImageView
18     lateinit var textView: TextView
19
20     override fun onCreate(savedInstanceState: Bundle?) {
21
22         super.onCreate(savedInstanceState)
23         setContentView(R.layout.activity_splash)
24
25         imgLogo = findViewById(R.id.imgLogo)
26         textView = findViewById(R.id.textView)
27
28         logo = AnimationUtils.loadAnimation(context: this,
29             R.anim.logo
```

```

super.onCreate(savedInstanceState)
setContentView(R.layout.activity_splash)

imgLogo = findViewById(R.id.imgLogo)
textView = findViewById(R.id.textView)

logo = AnimationUtils.loadAnimation(context: this,
    R.anim.Logo
)
name = AnimationUtils.loadAnimation(context: this,
    R.anim.name
)

imgLogo.startAnimation(logo)
textView.startAnimation(name)

Handler().postDelayed({
    val intent = Intent(packageContext: this, LoginActivity::class.java)
    startActivity(intent)
    finish()
}, delayMillis: 2200)

```

Activity > var logo

Version Control Build Logcat Terminal

4 commits // View Commits (23 minutes ago)

REGISTER THE USER

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help BookApp [C:\Users\app]

src > main > java > com > miniproject > bookapp > activity > RegisterActivity

HomeActivity.kt x ForgetPasswordActivity.kt x LoginActivity.kt x RegisterActivity.kt x SplashA

1 package com.miniproject.bookapp.activity

2

3 import android.app.AlertDialog

4 import android.content.Intent

5 import android.os.Bundle

6 import android.provider.Settings

7 import android.view.View

8 import android.widget.*

9 import androidx.appcompat.app.AppCompatActivity

10 import androidx.core.app.ActivityCompat

11 import com.google.android.material.textfield.TextInputEditText

12 import com.google.firebase.auth.FirebaseAuth

13 import com.google.firebase.firestore.FirebaseFirestore

14 import com.miniproject.bookapp.R

15 import com.miniproject.bookapp.model.User

16 import com.miniproject.bookapp.util.ConnectionManager

17

18

19 class RegisterActivity : AppCompatActivity() {

20

21 lateinit var edtname: EditText

22 lateinit var edtemail: EditText

23 lateinit var edtphone: EditText

24 lateinit var edtpassword: TextInputEditText

25 lateinit var edtconfirmpassword: TextInputEditText

26 lateinit var btnCreateAcc: Button

27 lateinit var progressBar: ProgressBar

28 lateinit var txtMember: TextView

29 lateinit var fauth: FirebaseAuth

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help BookApp [C:\Users\user\AndroidStudioProjects\Book_finder_app_mini_project] - ...activity\Reg
app | src | main | java | com | miniproject | bookapp | activity | RegisterActivity | No Devices | [Run] [Debug] [Test] [Profile] [Monitor] [Logcat] [Terminal] [Git]
HomeActivity.kt | ForgetPasswordActivity.kt | LoginActivity.kt | RegisterActivity.kt | SplashActivity.kt
31
32 override fun onCreate(savedInstanceState: Bundle?) {
33
34     super.onCreate(savedInstanceState)
35     setContentView(R.layout.activity_register)
36
37     init()
38
39     progressBar.visibility = View.GONE
40
41     btnCreateAcc.setOnClickListener { it: View!
42
43         val name = edtname.text.toString()
44         val email = edtemail.text.toString()
45         val phoneno = edtphone.text.toString()
46         val password = edtpassword.text.toString()
47         val confirmPassword = edtconfirmpassword.text.toString()
48
49         if (name.isNotEmpty() && email.isNotEmpty() && phoneno.isNotEmpty() && password.isNotEmpty() && confirmPassword.isNotEmpty()) {
50             if (password.equals(confirmPassword) && phoneno.length == 10) {
51                 if (ConnectionManager().checkConnectivity(context = this@RegisterActivity)) {
52
53                     progressBar.visibility = View.VISIBLE
54                     fauth.createUserWithEmailAndPassword(email, password)
55                         .addOnCompleteListener { task ->
56                             if (task.isSuccessful) {
57                                 val user = User(
58                                     name,
59                                     email,
```

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help BookApp [C:\Users\user\AndroidStudioProjects\Book_finder_a
app | src | main | java | com | miniproject | bookapp | activity | RegisterActivity | No Devices | [Run] [Debug] [Test] [Profile] [Monitor] [Logcat] [Terminal] [Git]
HomeActivity.kt | ForgetPasswordActivity.kt | LoginActivity.kt | RegisterActivity.kt | SplashActivity.kt
60
61     phoneno,
62     password
63 )
64
65     val userid = fauth.currentUser?.uid
66     if (userid != null) {
67         firestore.collection(collectionPath = "users").document(userid).set(user)
68         .addOnCompleteListener { task ->
69             if (task.isSuccessful) {
70                 startActivity(
71                     Intent(
72                         packageContext = this,
73                         HomeActivity::class.java
74                     )
75                 )
76                 finish()
77                 Toast.makeText(
78                     context = this,
79                     text = "Registered Successfully",
80                     Toast.LENGTH_SHORT
81                 ).show()
82             }
83         }
84     } else {
85         Toast.makeText(
86             context = this,
87             text = "Some error occurred",
88             Toast.LENGTH_SHORT
89         ).show()
```

```
147     }
148 }
149
150 private fun init() {
151     edtname = findViewById(R.id.edtname)
152     edtemail = findViewById(R.id.edtemail)
153     edtphone = findViewById(R.id.edtphone)
154     edtpassword = findViewById(R.id.edtpassword)
155     edtconfirmpassword = findViewById(R.id.edtconfirmpassword)
156     btnCreateAcc = findViewById(R.id.btnCreateAcc)
157     progressBar = findViewById(R.id.progressBar)
158     txtMember = findViewById(R.id.txtMember)
159     fauth = FirebaseAuth.getInstance()
160     fstore = FirebaseFirestore.getInstance()
161 }
162
163 private fun showError(inputField: EditText) {
164     if (inputField.text.isEmpty()) {
165         inputField.error = "Please fill out this field"
166     }
167 }
168
169 /*override fun onSupportNavigateUp(): Boolean {
170     onBackPressed()
171     return true
172 }*/
173 }
```

References

- Google Book API
- Google
- Wikipedia
- Android Development course from Internshala (for learning purpose only)
- Github repository link for the project:

https://github.com/JanviPangoriya/Book_finder_app_mini_project
