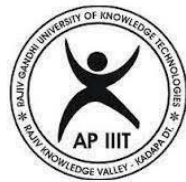


# “BOOKSCAPE”

## BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING



**RGUKT**

**Rajiv Gandhi University of Knowledge Technologies  
R.K.VALLEY**

**Submitted by**

**G V Jahnavi - R170075**

**M N V Bhavya Vani - R170081**

**P Abhinaya - R170754**

**Under the Esteemed guidance  
of Mr. Satyanandaram N  
RGUKT RK Valley.**

## **DECLARATION**

We hereby declare that the report of the B.Tech Minor Project Work entitled **“BOOKSCAPE”** which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for award of any degree.

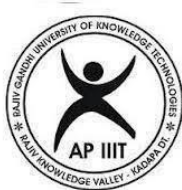
**G V Jahnavi – R170075**

**M Naga Venkata Bhavya Vani– R170081**

**P Abhinaya – R170754**

**Dept. Of Computer Science and Engineering.**

# RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES



**RGUKT**

(A.P.Government Act 18 of 2018)

RGUKT, RK VALLEY

Department of Computer Science and Engineering

## CERTIFICATE FOR PROJECT COMPLETION

This is to certify that the project entitled “**BOOKSCAPE**” submitted by **G.V. Jahnavi (R170075), M.N.V. Bhavya Vani (R170081), P. Abhinaya (R170754)** under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering during the third year academic semester-2 2021-2022 at RGUKT, RK VALLEY. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

### **Project Internal Guide**

Mr.N.Satyanandaram  
Assistant Professor  
RGUKT,RK Valley.

### **Head of the Department**

Mr.P. Harinadha  
HOD of CSE  
RGUKT, RK Valley.

## **ABSTRACT**

**“Bookscape”** is an Android based Mobile application for reading books online. This application allows users to read books from various categories like fiction to non-fiction without any paper cuts. This application is completely user friendly and secure. It provides free books to read with an aim of “ making knowledge accessible to every avid reader ” .This app is designed for bibliophiles who could never find enough books to quench their reading desire. This app allows user to read books conveniently in any category of their choice at any time and place. Admin of the app can organise the books efficiently according to their genre and also upload the new releases accordingly. It is tedious to carry each and every book you want to read at a time with you instead you can choose and read any book from best selling authors around the world with this app. You will be just one click away from a fantastical reading experience!!

## **Index**

1. Abstraction	4
2.Introduction	6
2.1 Purpose	6
2.2 Intended Audience	6
2.3 Product Vision	6
2.4 Technologies	6
3. Android Studio	7
4. Firebase	8
5. System in Context	9
5.1 Context Diagram	10
6. System-Wide Requirements	10
7. Function Requirements	12
7.1 Use Case Diagram	12
7.2 Use Case Overview	13
7.3 Use Case Specification	13
8.ER Diagram	18
9. Implementation and System Testing	19
10. Source code	20
11. Result	36
12. References	42

# **BOOKSCAPE SRS DOCUMENT**

## **Introduction:**

This document has the requirements of Bookscape. The Bookscape app is used to provide free books for readers.

## **1.1: Purpose**

The purpose of this document is to gather the requirements that are needed for implementing the Bookscape . It also focuses on various key features, the product, product vision and scope, product overview. The main purpose of Bookscape is to provide a platform to the user to read books freely and easily .

## **1.2: Intended Audience:**

The intended audience will be the users who can read the books from various categories which are uploaded by Admin.

## **Users:**

1. Reader
2. Admin

## **Product Vision:**

### **Vision Statement:**

The product vision is to cater to various readers with books that are freely available. This product is user friendly and easily accessible. Bookscape provides books from various genres with free of cost and the user can read at anytime and anywhere.

## **Technologies:**

- Android Studio
- Firebase

## **Android Studio:**

Android Studio provides many excellent features that enhance productivity when building Android apps, such as a blended environment where one can develop for all Android devices, apply Changes to push code and resource changes to the running app without restarting the app, a flexible Gradlebased build system, a fast and feature-rich emulator, GitHub and Code template integration to assist you to develop common app features and import sample code, extensive testing tools and frameworks, C++ and NDK support, and many more.

### **Features:**

**The following features are provided in the current stable version.**

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layout on multiple screen configurations.
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

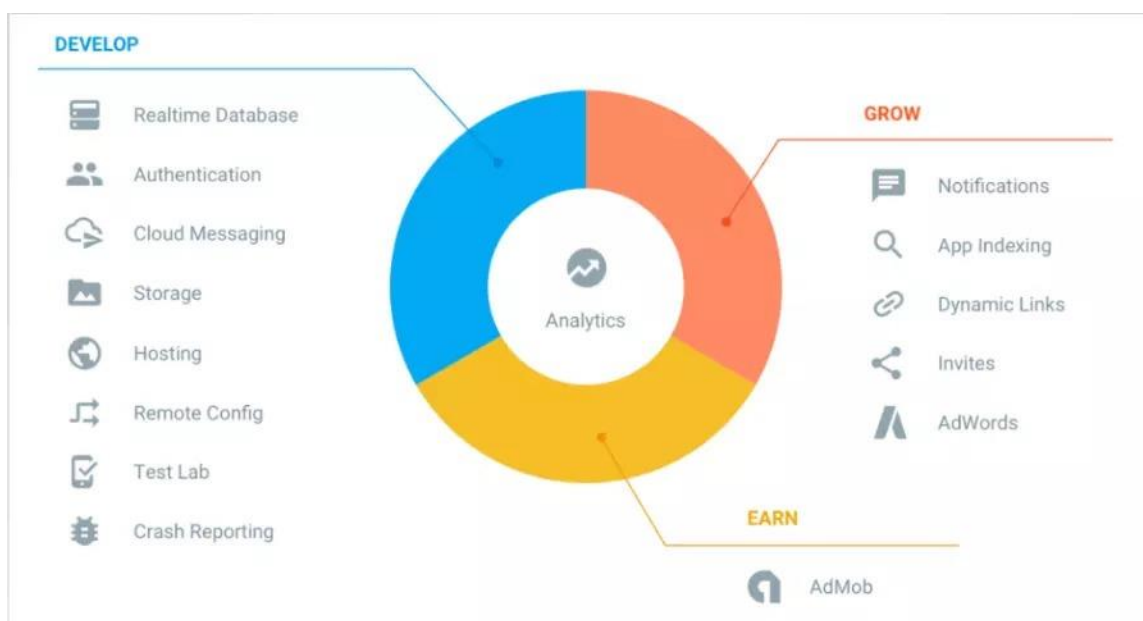
Android Studio supports all the same programming languages of IntelliJ (and CLion e.g. Java, C++, and more with extensions, such as Go and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features.s While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.Once an app has been compiled with Android Studio, it can be published on the Google Play Store. The application has to be in line with the Google Play Store developer content policy.

## Firestore:

Firestore is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firestore side which makes it easy to use its features more efficiently. It provides services to android, ios, web, and unity. It provides cloud storage. It uses NoSQL for the database for the storage of data.

## Features of Firestore:

Mainly there are 3 categories in which firestore provides its services.

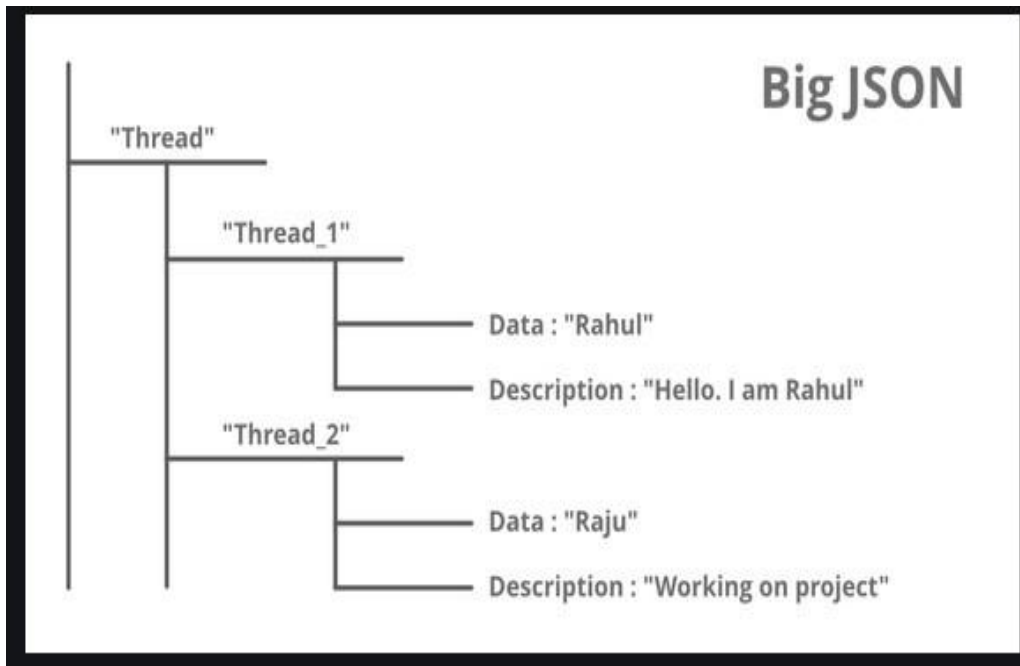


### ***Build better applications***

This feature mainly includes backend services that help developers to build and manage their applications in a better way. Services included under this feature are :

- **Realtime Database:** The Firebase Realtime Database is a cloud-based NoSQL database that manages your data at the blazing speed of milliseconds. In simplest term, it can be considered as a big JSON file.
- **Cloud Firestore:** The cloud Firestore is a NoSQL document database that provides services like store, sync, and query through the application on a global scale. It stores data in the form of objects also known as Documents. It has a key-value pair and can store all kinds of data like, strings, binary data, and even JSON trees.



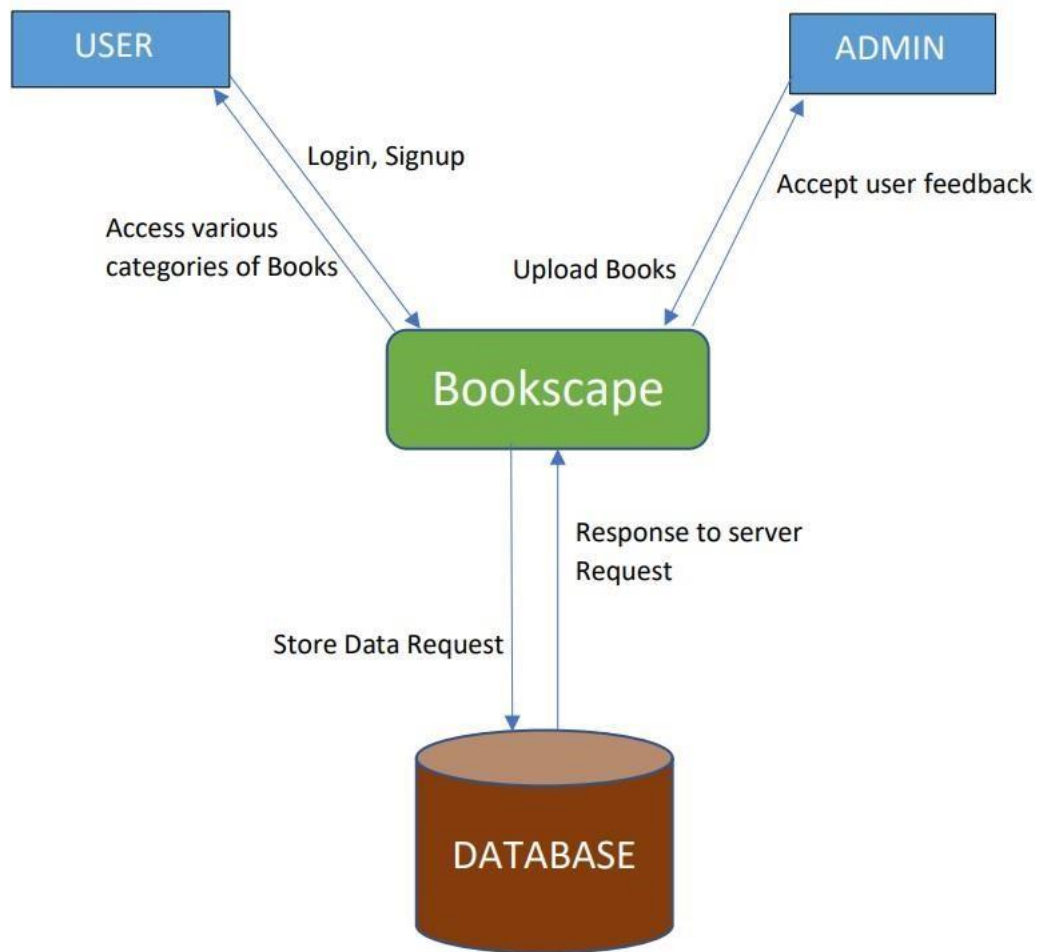


- **Authentication:** Firebase Authentication service provides easy to use UI libraries and SDKs to authenticate users to your app. It reduces the manpower and effort required to develop and maintain the user authentication service. It even handles tasks like merging accounts, which if done manually can be hectic.

### System in Context:

The Bookscape provides different categories of books to the reader. User can see the recent releases of books which are updated by admin and can read books at anytime and anywhere. This app reduces the burden of carrying books by simply read any book with a single touch.

### Context Diagram:



### System-wide Requirements(Received):

#### Actors:

This system interacts with two kinds of users. Each user has its own functionalities to access with the system. The functionalities of users are independent of each other.

#### Events:

Bookscape is a Multi-user system which provides activities associated with reading books.

#### The most critical events are:

1. Users gets register using the EmailId ,Name,Contact Number.
2. User can Login using the email id and password.
3. User can see list of books.

4.User can see various kinds of books by selecting desired category.

5.User can edit profile and can change the user details.

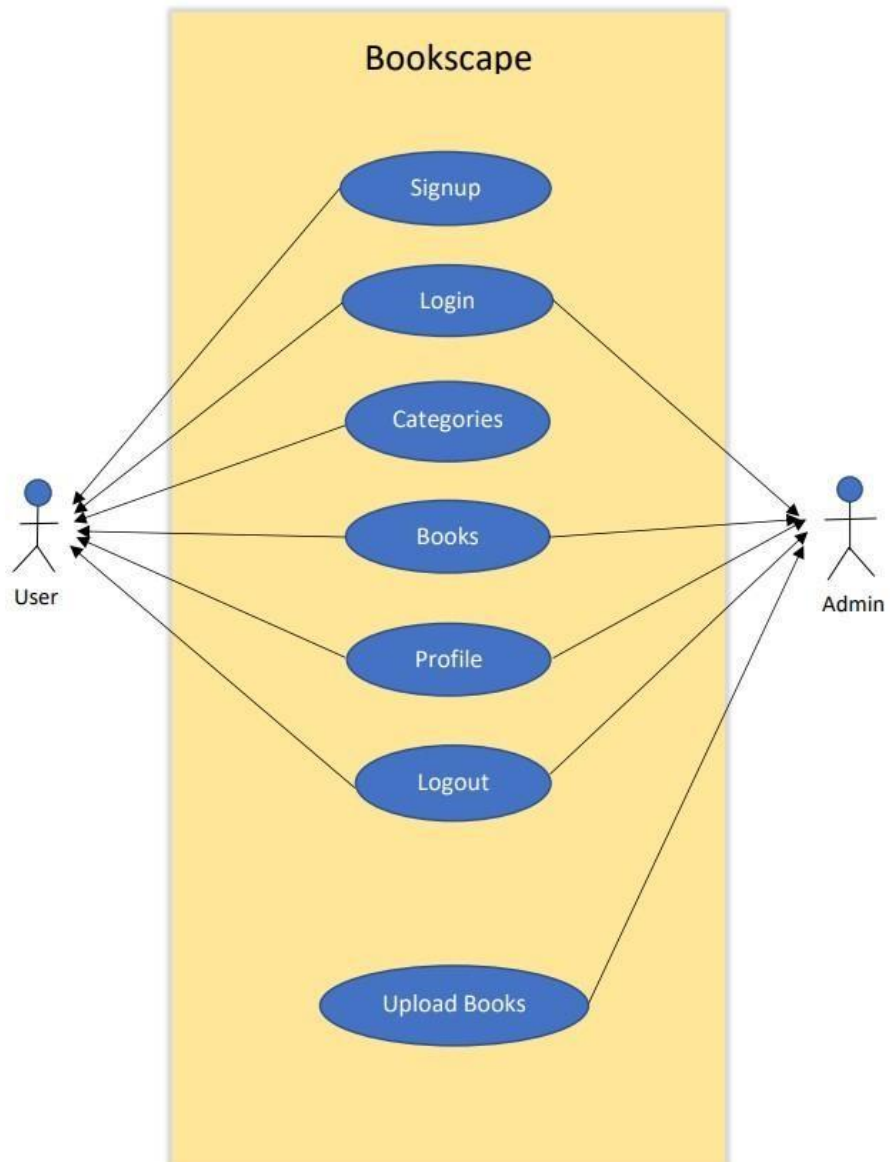
6.User can change their Login Credentials.

**The below table provides a set of user visible events that define the functionalities that are in Bookscape.**

	Actor	Action	Object	Frequency	Arrival Pattern	Response
1	User	OnClick	Signup	1/day	Episodic	It asks Name,email address,contact,create password for Registration
2	User	OnClick	Login	1/day	Episodic	It asks email address and password for Login
3	User	OnClick	Book	1/day	Episodic	It opens the pdf for the selected particular book.
4	User	OnClick	Categories	1/day	Episodic	It opens the list of categories.
5	User	OnClick	Category	1/day	Episodic	It opens the list of books for the particular category.
6	User	OnClick	Profile	1/day	Episodic	It shows the user details.
7	User	OnClick	Edit Profile	1/day	Episodic	It allows the user to edit the details.
8	User	OnClick	Change Password	1/day	Episodic	It allows the users to change the password.
9	User	OnClick	Logout	1/day	Episodic	It logout from the account.
10	User	OnClick	Forgot Password	1/day	Episodic	It sends the reset password link to the user email id.
11	User	OnClick	Send Feedback	1/day	Episodic	User can send feedback about the app.
12	Admin	OnClick	Login	1/day	Episodic	It asks email address and password for Login
13	Admin	OnClick	Upload Books	1/day	Episodic	Here admin can upload the books into application.

## Functional Requirements :

### Use case Diagram:



## Use case Overview :

S.No	Use-case Id	Use-case Name	Priority	Stability	Verifiable
1	UC-BS-S	Signup	High	Stable	Verifiable
2	UC-BS-LI	Login	High	Stable	Verifiable
3	UC-BS-LOB	List Of Books	High	Not Stable	Verifiable
4	UC-BS-C	Categories	High	Stable	Verifiable
5	UC-BS-Ca	Category	High	Stable	Verifiable
6	UC-BS-B	Book	High	Stable	Verifiable
7	UC-BS-P	Profile	High	Stable	Verifiable
8	UC-BS-LO	Logout	High	Stable	Verifiable
9	UC-BS-U	Upload Books	High	Not Stable	Verifiable

## Use-Case Specification:

### 1.UC-BS-S : Signup

Use-case Id : UC-BS-S	
Description	User clicks the Register button
Preconditions	1.User must install&open Bookscape App. 2.User must click on Get Started button
Success guarantee (post conditions)	After the user registered he/she will have an account then it redirects to Home page.
Frequency of use	High
Main success scenario( basic flow)	1.User access the Bookscape App. 2.User enters details. 3.Click on register. 4.Redirects to Home page.
Extensions (alternate flows)	If the user didn't enter user details then he/she can't enter into the app.
Frequency of occurrence	Very high

## 2. UC-BS-LI : Login

Use-case Id :UC-BS-LI	
Description	User enter login credentials clicks the Login button
Preconditions	1.User must open Bookscape App. 2.User must register an account.
Success guarantee (post conditions)	After the user clicks on login button it redirects to Home page.
Frequency of use	High
Main success scenario ( basic flow)	1.User access the Bookscape App. 2.User enters login credentials clicks on login button 3.Redirects to Home page.
Extensions (alternate flow)	If the user didn't enter login credentials then he/she can't enter into the app.
Frequency of occurrence	Very high

## 3. UC-BS-C : List of books

Use-case Id : UC-BS-LOB	
Description	Home page displays list of books ,user can access book by clicking it.
Preconditions	1.User must Login into app.
Success guarantee (post conditions)	After the user click on the book he can read book.
Frequency of use	High
Main success scenario ( basic flow)	1.Click on Book. 2.Opens book in pdf format.
Extensions (alternate flow)	If the user didn't click on book then pdf won't open.
Frequency of occurrence	Very high

#### 4. UC-BS-C : Categories

Use-case Id : UC-BS-C	
Description	User clicks on the categories icon to get the list of categories.
Preconditions	1.User must click on categories icon .
Success guarantee (post conditions)	After the user click on the categories then it redirects to list of categories page.
Frequency of use	High
Main success scenario ( basic flow)	1.Click on Categories Icon. 2.Redirects to list of categories page.
Extensions (alternate flow)	If the user didn't click on the categories then he/she didn't redirects to list of categories page.
Frequency of occurrence	Very high

#### 5. UC-BS-Ca : Category

Use-case Id : UC-BS-Ca	
Description	User clicks on the category it shows the list of books of a particular category.
Preconditions	1.User must open a particular category.
Success guarantee (post conditions)	After the user click on the category then it displays the list of books of particular category.
Frequency of use	High
Main success scenario ( basic flow)	1.Click on Category. 2.Displays the list of books.
Extensions (alternate flow)	If the user didn't click on the category then it didn't show the list of books
Frequency of occurrence	Very high

## 6. UC-BS-B : Book

Use-case Id : UC-BS-B	
Description	User clicks on the book it opens the pdf of the book.
Preconditions	1.User must open a particular book.
Success guarantee (post conditions)	After the user click on the book then it displays pdf of particular book.
Frequency of use	High
Main success scenario ( basic flow)	1.Click on Book. 2.Displays the book in pdf format.
Extensions (alternate flow)	If the user didn't click on the book then he can't read the book
Frequency of occurrence	Very high

## 7. UC-BS-P : Profile

Use-case Id : UC-BS-P	
Description	User clicks on Profile icon, it will displays user details.
Preconditions	1.User must clicks on Profile icon.
Success guarantee (post conditions)	After the user click on the Profile then it displays the details of user
Frequency of use	High
Main success scenario ( basic flow)	1.Click on Profile icon. 2.Displays details of user
Extensions (alternate flow)	If the user didn't click on profile it won't display user details.
Frequency of occurrence	Very high



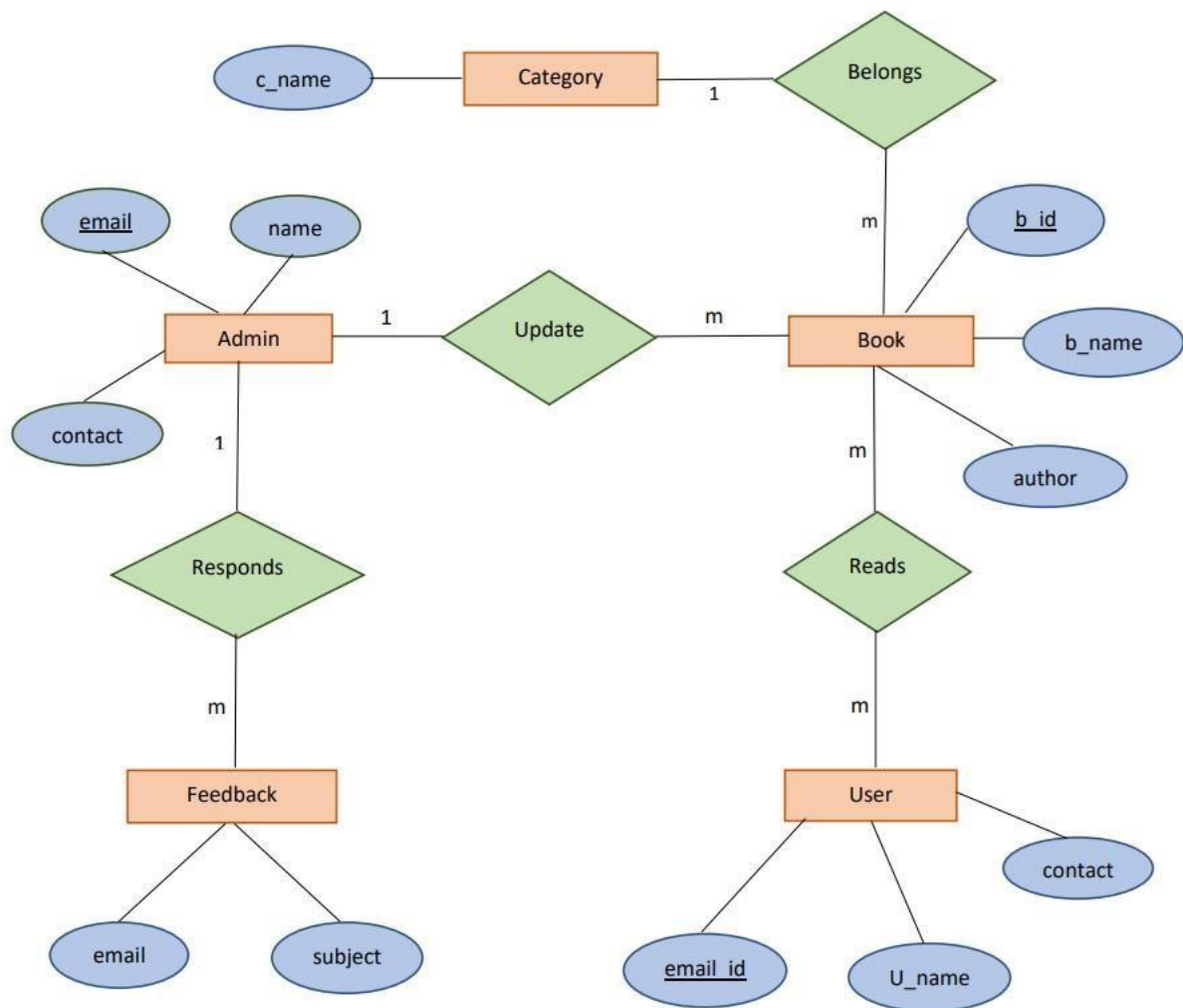
## 8. UC-BS-LO : Logout

Use-case Id : UC-BS-LO	
Description	User clicks on Logout, it will logout from the account
Preconditions	1.User must clicks on Logout Button.
Success guarantee (post conditions)	After the user click on the Logout then it redirects to Login page
Frequency of use	High
Main success scenario (basic flow)	1.Click on Logout Button. 2.Logout from the account.
Extensions (alternate flow)	If the user didn't click on Logout it won't redirect to Login page
Frequency of occurrence	Very high

## 9. UC-BS-U : Upload

Use-case Id : UC-BS-U	
Description	Admin clicks on Upload to upload books into the Application.
Preconditions	1.Admin must login into app with credentials.
Success guarantee (post conditions)	Once admin upload the book it will be stored in Firebase Storage
Frequency of use	High
Main success scenario (basic flow)	1.Click on Upload Button. 2.Book will upload into Firebase.
Extensions (alternate flow)	If the user didn't click on upload then he can't upload book.
Frequency of occurrence	Very high

## ER Diagram :



## **Implementation and System Testing**

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

### **System Testing**

The goal of the system testing process was to determine all faults in our project .The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing
- 2 .Integration testing

### **Unit Testing**

Unit testing is commenced when a unit has been created and effectively reviewed .In order to test a single module we need to provide a complete environment i.e. besides the section we would require The procedures belonging to other units that the unit under test calls Non local data structures that module accesses .A procedure to call the functions of the unit under test with appropriate parameters

#### **1. Test for the admin module**

Testing admin login form-This form is used for log in of administrator of the system. In this form we enter the Email Id and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask the details.

### **Integration Testing**

In the Integration testing we test various combination of the project module by providing the input. The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

## SOURCE CODE :

### 1.REGISTER

#### Activity\_registration.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".RegistrationPage">

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="373dp"
        android:layout_height="558dp"
        android:layout_marginStart="60dp"
        android:layout_marginTop="120dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="30dp"
        tools:ignore="ImageContrastCheck" />

    <TextView
        android:id="@+id/registerTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="70dp"
        android:fontFamily="sans-serif-medium"/>

    <ScrollView
        android:id="@+id/scrollView2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="20dp"
        android:layout_marginTop="160dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="45dp"
        app:layout_constraintTop_toTopOf="parent" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <EditText
                android:id="@+id/fullnameEdittext"
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:layout_marginStart="10dp"
```

```

        android:hint="Name"
        android:inputType="textPersonName"
        android:padding="10dp"
        android:textSize="20sp"/>

<EditText
    android:id="@+id/ContactEdittext"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:layout_marginStart="10dp"
    android:layout_marginTop="20dp"
    android:layout_marginEnd="10dp"
    android:textSize="20sp" />

<EditText
    android:id="@+id/emailEdittext"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:hint="Email"
    android:inputType="textEmailAddress"
    android:padding="10dp"
    android:textSize="20sp"/>

<EditText
    android:id="@+id/setPasswordEdittext"
    android:layout_width="match_parent"
    android:hint="Set Password"
    android:inputType="textPassword"
    android:padding="10dp"
    android:textSize="20sp"
    android:visibility="visible" />

<EditText
    android:id="@+id/reenterPasswordEdittext"
    android:layout_width="match_parent"
    android:hint="Re-enter Password"
    android:inputType="textPassword"
    android:padding="10dp"
    android:textSize="20sp" />

<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="17dp"
    android:layout_marginRight="17dp" />

<Button
    android:id="@+id/registerButton"
    android:layout_width="120dp"
    android:layout_height="50dp"

```

```

        android:background="@drawable/background_color"
        android:text="Register"
        android:textColor="#FFFFFF"
        android:textSize="20sp"/>

```

```

    </LinearLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

## Registration.java

```

package com.android.goeasy;
import ...;

public class RegistrationPage extends AppCompatActivity {
    Button button;
    EditText name, contact, setPassword, reenterPassword, email;
    TextView textView;
    String emailPattern = "[a-zA-Z0-9._-]+@[a-z]+\\.[a-z]+";
    String contactPattern = "^([0-9]{10})$";

    FirebaseAuth mAuth;
    FirebaseUser mUser;
    FirebaseDatabase rootNode;
    DatabaseReference reference;
    ProgressDialog progressDialog;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration_page);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);

        button = findViewById(R.id.registerButton);
        setPassword = findViewById(R.id.setPasswordEditText);
        reenterPassword = findViewById(R.id.reenterPasswordEditText);
        textView = findViewById(R.id.textView);
        name = findViewById(R.id.fullnameEditText);
        contact = findViewById(R.id.ContactEditText);
        email = findViewById(R.id.emailEditText);
        progressDialog = new ProgressDialog(this);
        mAuth = FirebaseAuth.getInstance();
        mUser = mAuth.getCurrentUser();

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                PerforAuth();
            }
        });
    }
}

```

```

}
private void PerforAuth() {
    String Email = email.getText().toString();
    String Name = name.getText().toString();
    String Contact = contact.getText().toString();
    String Password = setPassword.getText().toString();
    String Resetpassword = reenterPassword.getText().toString();
    if (!Email.matches(emailPattern)) {
        email.setError("Enter correct email");
    } else if (Contact.length() != 10) {
        contact.setError("Enter valid contact number");
    } else if (Password.isEmpty() || Password.length() < 6) {
        setPassword.setError("Enter correct Password");
    } else if (!Password.equals(Resetpassword)) {
        reenterPassword.setError("Password not matched both field");
    } else if (Name.isEmpty()) {
        name.setError("Enter name");
    } else {
        progressDialog.setMessage("Please wait while Registration...");
        progressDialog.setTitle("Registration");
        progressDialog.setCanceledOnTouchOutside(false);
        progressDialog.show();

        mAuth.createUserWithEmailAndPassword(Email, Password).addOnCompleteListener(new
        OnCompleteListener < AuthResult > () {
            @Override
            public void onComplete(@NonNull Task < AuthResult > task) {
                if (task.isSuccessful()) {
                    UserHelper user = new UserHelper(Name, Email, Contact, Password);
                    FirebaseDatabase.getInstance()
                        .getReference("Users").child(FirebaseAuth.getInstance()
                            .getCurrentUser().getUid()).setValue(user)
                        .addOnCompleteListener(new OnCompleteListener < Void > () {
                            @Override
                            public void onComplete(@NonNull Task < Void > task) {
                                if (task.isSuccessful()) {
                                    Toast.makeText(RegistrationPage.this, "Registration succesful",
                                    Toast.LENGTH_SHORT).show(); } } });
                                progressDialog.dismiss();
                                sendUserToNextActivity();
                            } else {
                                progressDialog.dismiss();
                                Toast.makeText(RegistrationPage.this, "" + task.getException(),
                                Toast.LENGTH_SHORT).show(); } } });
                            private void sendUserToNextActivity() {
                                Intent intent = new Intent(RegistrationPage.this, HomePage.class);
                                intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
                                Intent.FLAG_ACTIVITY_NEW_TASK);
                                startActivity(intent); } }

```

## HOME PAGE:

### activity\_home\_page.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:isScrollContainer="true"
    android:nestedScrollingEnabled="true"
    android:orientation="vertical"
    android:overScrollMode="always"
    android:scrollbars="vertical"
    app:layout_scrollEffect="compress"
    app:layout_scrollFlags="scroll"
    tools:context=".HomePage">
    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/background_color">
        <RelativeLayout
            android:id="@+id/relative_layout"
            android:layout_width="match_parent"
            android:layout_height="95dp">
            <TextView
                android:layout_width="270dp"
                android:layout_height="wrap_content"
                android:layout_marginStart="10dp"
                android:text="Bookscape"
                android:textColor="@color/white"
                android:textSize="35dp">
            </TextView>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="10dp"
                android:layout_marginTop="60dp"
                android:text="escape to new world"
                android:textColor="@color/white"
                android:textSize="20dp">
            </TextView>
            <ImageView
                android:id="@+id/menu"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="350dp"
                android:layout_marginTop="23dp"
                android:src="@drawable/ic_baseline_menu_24"/>
        </RelativeLayout>
```



```

</com.google.android.material.appbar.AppBarLayout>
<ScrollView
    android:id="@+id/scrollView"
    android:layout_width="match_parent"
    android:layout_height="604dp">
    <RelativeLayout
        android:id="@+id/home_relative_layout"
        android:layout_width="match_parent"
        android:layout_height="565dp"
        android:orientation="vertical">
        <com.smarteist.autoimageslider.SliderView
            android:id="@+id/image_slider"
            android:layout_width="match_parent"
            android:layout_height="174dp"
            android:layout_alignParentTop="true"
            android:layout_alignParentEnd="true"
            app:sliderAnimationDuration="600"
            app:sliderAutoCycleDirection="back_and_forth"
            app:sliderIndicatorAnimationDuration="600"
            app:sliderIndicatorEnabled="true"
            app:sliderIndicatorGravity="center_horizontal|bottom"
            app:sliderIndicatorMargin="15dp"
            app:sliderIndicatorOrientation="horizontal"
            app:sliderIndicatorPadding="3dp"
            app:sliderIndicatorRadius="2dp"
            app:sliderIndicatorSelectedColor="#5A5A5A"
            app:sliderIndicatorUnselectedColor="#FFF"
            app:sliderScrollTimeInSec="2"
            app:sliderStartAutoCycle="true"/>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="2dp"
            android:layout_marginTop="190dp"
            android:text="@string/new_releases"
            android:textColor="@color/black"
            android:textSize="20sp">
        </TextView>
        <HorizontalScrollView
            android:id="@+id/horizontal"
            android:layout_width="match_parent"
            android:layout_height="190dp"
            android:layout_marginTop="230dp"
            android:scrollbars="none"
            tools:ignore="SpeakableTextPresentCheck">
            <LinearLayout
                android:id="@+id/gallery"
                android:layout_width="wrap_content"
                android:layout_height="190dp"
                android:orientation="horizontal"/>
        </HorizontalScrollView>

```

```

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/recyclerView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentEnd="true"
            android:layout_marginTop="430dp"
            android:layout_marginEnd="0dp"
            android:background="@color/white">
        </androidx.recyclerview.widget.RecyclerView>
    </RelativeLayout>
</ScrollView>
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottomNavigationView"
    android:layout_width="match_parent"
    android:layout_height="55dp"
    android:background="@drawable/background_color"
    app:itemIconSize="50dp"
    app:itemIconTint="@color/white"
    app:itemTextColor="#FFFFFF"
    app:labelVisibilityMode="unlabeled"
    app:menu="@menu/bottom_nav_menu">
</com.google.android.material.bottomnavigation.BottomNavigationView>
</LinearLayout>

```

### **HomePage.java**

```

package com.android.goeasy;
import ...;
public class HomePage extends AppCompatActivity {
    SliderView sliderView;
    RecyclerView recyclerView;
    ImageView imageView, imageView2;
    HorizontalScrollView horizontalScrollView;
    FirebaseDatabase firebaseDatabase;
    DatabaseReference databaseReference;
    FirebaseAuth mAuth;
    FirebaseUser mUser;
    RelativeLayout relativeLayout;
    List < Book > bookList;
    CardView cardView;
    int[] sliderImages = {
R.drawable.image1, R.drawable.image2, R.drawable.image3, R.drawable.image4,R.drawable.image5
    };
    int[] books = {
        R.drawable.book1, R.drawable.book2, R.drawable.book3, R.drawable.book4, R.drawable.book5,
R.drawable.book6, R.drawable.book7, R.drawable.book8};
    String[] booksnames = {
        "Percy Jackson and the Greek Heroes", "Glass Sword", "Days of Blood and Starlight", "The
Bane Chronicles", "Lord of Shadows", "The Lost Hero", "The Battle of Labyrinth", "Magnus
chase and the Gods of Asgard" };
    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home_page);
    sliderView = findViewById(R.id.image_slider);
    imageView = findViewById(R.id.menu);
    imageView2 = findViewById(R.id.card_image1);
    SliderAdapter sliderAdapter = new SliderAdapter((sliderImages));
    sliderView.setSliderAdapter(sliderAdapter);
    sliderView.setIndicatorAnimation(IndicatorAnimationType.WORM);
    sliderView.setSliderTransformAnimation(SliderAnimations.DEPTHTRANSFORMATION);
    sliderView.setScrollTimeInSec(3);
    sliderView.startAutoCycle();
    mAuth = FirebaseAuth.getInstance();
    mUser = mAuth.getCurrentUser();
    cardView = findViewById(R.id.cardview_imageview);
    relativeLayout = findViewById(R.id.recycle_relative);
    firebaseDatabase = FirebaseDatabase.getInstance();
    databaseReference = firebaseDatabase.getReference("Books");
    LinearLayout gallery = findViewById(R.id.gallery);
    LayoutInflater inflater = LayoutInflater.from(this);
    horizontalScrollView = findViewById(R.id.horizontal);
    bookList = new ArrayList < > ();
    for (int i = 0; i < 8; i++) {
        View view = inflater.inflate(R.layout.gallery_item, gallery, false);
        ImageView imageView = view.findViewById(R.id.galleryimageView);
        imageView.setImageResource(books[i]);
        TextView textView = view.findViewById(R.id.gallerytextView);
        textView.setText(booksnames[i]);
        gallery.addView(view); }
    recyclerView = findViewById(R.id.recyclerView);
    recyclerView.setLayoutManager(new LinearLayoutManager(HomePage.this));
    registerForContextMenu(imageView);
    bookList.add(new Book("Lethal White", "J.K.Rowling", R.drawable.lethal_white,
"https://firebasestorage.googleapis.com/v0/b/janu-go-easy.appspot.com/o/Mystery%20F4.Lethal%
20White.pdf?alt=media&token=547d90da-cdac-477c-b114-9a222fe15cb5"));
    bookList.add(new Book("Troubled Blood", "J.K.Rowling", R.drawable.troubled_blood,
"https://firebasestorage.googleapis.com/v0/b/janu-go-
easy.appspot.com/o/Mystery%20F5.Troubled%20Blood.pdf?alt=media&token=
e3ea9385-e0ab-4988-818e-bf03469cf514"));
    bookList.add(new Book("Seven Days", "Alex Lake", R.drawable.seven_days,
"https://firebasestorage.googleapis.com/v0/b/janu-go-
easy.appspot.com/o/Mystery%20FFAMOUS%20FIVE%20COMPLETE%20SERIES.pdf?alt=media&to
ken=3c5426bf-c60e-4e88bc03-f67a999e7910")); bookList.add(new Book("The Half Blood Prince",
"J.K.Rowling", R.drawable.hp6, "https://firebasestorage.googleapis.com/v0/b/janu-go-

```

```

        easy.appspot.com / o / Fantasy % 2 F6.The % 20 Half % 20 Blood % 20 Prince.pdf ? alt =
media & token = d80037a1 - 8e7 e - 4941 - ba2d - cbf100477c9a ");    bookList.add(new Book("
        The Deathly Hallows ", "
        J.K.Rowling ", R.drawable.hp7, "
        https : //firebasestorage.googleapis.com/v0/b/janu-go-
easy.appspot.com/o/Fantasy%2F7.%20The%20Deathly%20Hallows.pdf?alt=media&token=96c8512f-
9e3a-4f05-8d26cb6ce37db3c0"));    bookList.add(new Book("HarryPotter and the cursed
child", "J.K.Rowling", R.drawable.cursed_child, "https://firebasestorage.googleapis.com/v0/b/janu-
goeasy.appspot.com/o/Fantasy%2FHarry-Potter-and-the-Cursed-
Child.pdf?alt=media&token=60f120ff-cbf1-4cb8-841f-0accf3c91cf9"));
        bookList.add(new Book("Alice in Wonderland", "Lewis Carrol",
R.drawable.enchantment_of_ravens, "https://firebasestorage.googleapis.com/v0/b/janugoeasy.appspot.c
om/o/Fantasy%2FAn%20Enchantment%20of%20Ravens-
Margaret%20Rogerson.pdf?alt=media&token=66e21e53-c146-434c-80e5-7482907ebcf1"));
        bookList.add(new Book("Sorcery of Thorns", "Margaret Rogerson", R.drawable.sorcery_of_thorns,
"https://firebasestorage.googleapis.com/v0/b/janugoeasy.appspot.com/o/Fantasy%2FSorcery%20of%2
0Thorns%20Margaret%20Rogerson.pdf?alt=media&token=dccf86bc-d8bd-4e32-a8f0-
66ebe92af187")); bookList.add(new Book("Stardust", "Neil Gaiman", R.drawable.stardust,
"https://firebasestorage.googleapis.com/v0/b/janu-go-easy.appspot.com/o/Fantasy%2FStardust-
Neil%20Gaiman.pdf?alt=media&token=ffd4955c-eb4d-4cd3-92c 3 b7cee01c34c "));
        ListAdapter adapter = new ListAdapter(this, bookList);
recyclerView.setAdapter(adapter);}

```

```

        @Override public void onCreateContextMenu(ContextMenu menu, View
v, ContextMenu.ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        MenuInflater menuInflater = getMenuInflater();
        menuInflater.inflate(R.menu.homepage_menu, menu);
        }

```

```

        @Override public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
        case R.id.about:
            startActivity(new Intent(HomePage.this, About.class));
            return true;
        case R.id.feedback:
            startActivity(new Intent(HomePage.this, Feedback.class));
            return true;
        default:
            return super.onOptionsItemSelected(item);} }

```

```

        public void onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
        case R.id.home:
            startActivity(new Intent(HomePage.this, HomePage.class));
            break;
        case R.id.category:
            startActivity(new Intent(HomePage.this, Categories.class));

```

```

        break;
    case R.id.profile:
        startActivity(new Intent(HomePage.this, Profile.class));
        break;}}}

```

### **activity\_categories.xml**

```

<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Categories">
    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/background_color">
        <RelativeLayout
            android:id="@+id/relative_layout"
            android:layout_width="wrap_content"
            android:layout_height="53dp">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="50dp"
            android:layout_marginStart="10dp"
            android:layout_marginTop="3dp"
            android:layout_marginBottom="2dp"
            android:fontFamily="sans-serif-black"
            android:text="Bookscape"
            android:textColor="@color/white"
            android:textSize="30sp">
        </TextView>
    </RelativeLayout>
</com.google.android.material.appbar.AppBarLayout>
<ImageView
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:src="@drawable/categ"
    android:scaleType="centerCrop">
</ImageView>
<GridView
    android:layout_width="match_parent"
    android:layout_height="545dp"
    android:id="@+id/categories_gridView"
    android:paddingTop="10dp"
    android:gravity="center"

```

```

        android:numColumns="2">
</GridView>
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottomNavigationView"
    android:layout_width="match_parent"
    android:layout_height="55dp"
    android:background="@drawable/background_color"
    app:itemBackground="@drawable/background_color"
    app:itemIconSize="50dp"
    app:itemIconTint="@color/white"
    app:itemTextColor="#FFFFFF"
    app:labelVisibilityMode="unlabeled"
    app:menu="@menu/bottom_nav_menu">
</com.google.android.material.bottomnavigation.BottomNavigationView> </LinearLayout>

```

### **Categories.java**

```

package com.android.goeasy;
import ...;
public class Categories extends AppCompatActivity {
    GridView gridView;
    ImageView imageView;
    TextView textView;
    ArrayList < String > text = new ArrayList < > ();
    ArrayList < Integer > image = new ArrayList < > ();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_categories);
        imageView = findViewById(R.id.grid_imageView);
        textView = findViewById(R.id.grid_textView);
        gridView = findViewById(R.id.categories_gridView);
        fillArray();
        GridAdapter gridAdapter = new GridAdapter(this, text, image);
        gridView.setAdapter(gridAdapter);
        gridView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView < ? > adapterView, View view, int i, long l) {
                Intent intent = new Intent(Categories.this, BooksList.class);
                String cate = text.get(i);
                intent.putExtra("cat", cate);
                startActivity(intent);} });}
    private void fillArray() {
        text.add("Dystopian");
        text.add("Fantasy");
        text.add("Mystery");
        text.add("Romance");
        text.add("Science Fiction");
        text.add("Young Adult");
    }
}

```

```

text.add("Ghost");
text.add("History");
text.add("Philosophy");
text.add("Poetry");
text.add("Epic");
text.add("Adventure");
image.add(R.drawable.dystopian);
image.add(R.drawable.fantasy);
image.add(R.drawable.mystery);
image.add(R.drawable.romance);
image.add(R.drawable.science_fiction);
image.add(R.drawable.young_adult);
image.add(R.drawable.ghost);
image.add(R.drawable.history);
image.add(R.drawable.philosophy);
image.add(R.drawable.poetry);
image.add(R.drawable.epic);
image.add(R.drawable.adventure)}
public void onMenuItemClick(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.home:
            startActivity(new Intent(Categories.this, HomePage.class));
            break;
        case R.id.category:
            startActivity(new Intent(Categories.this, Categories.class));
            break;
        case R.id.profile:
            startActivity(new Intent(Categories.this, Profile.class));
            break;
    }
}

```

#### **activity\_books\_list.xml**

```

<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".BooksList">
    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/background_color">
        <RelativeLayout
            android:id="@+id/relative_layout"
            android:layout_width="wrap_content"
            android:layout_height="50dp">

```

```

<TextView
    android:id="@+id/back_to_categories"
    android:layout_width="57dp"
    android:layout_height="48dp"
    android:layout_marginTop="3dp"
    android:layout_marginBottom="2dp"
    android:clickable="true"
    android:fontFamily="sans-serif-black"
    android:textColor="@color/white"
    app:drawableRightCompat="@drawable/ic_launcher_back_button_foreground"
tools:ignore="SpeakableTextPresentCheck,SpeakableTextPresentCheck,TouchTargetSizeCheck,Touch
TargetSizeCheck,TouchTargetSizeCheck">
</TextView>

```

```

</RelativeLayout>
</com.google.android.material.appbar.AppBarLayout>
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/books_list_recycler_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</androidx.recyclerview.widget.RecyclerView>
</LinearLayout>

```

### **BooksList.java**

```

package com.android.goeasy;
import ...;
public class BooksList extends AppCompatActivity {
    RecyclerView recyclerView;
    TextView textView;
    List < Book > bookList;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_books_list);
        recyclerView = findViewById(R.id.books_list_recycler_view);
        recyclerView.setLayoutManager(new LinearLayoutManager(BooksList.this));
        Bundle bundle = getIntent().getExtras();
        String category = bundle.getString("cat");
        bookList = new ArrayList < > ();
        if (category.equals("Dystopian")) { ... }
        if (category.equals("Fantasy")) { ... }
        if (category.equals("Mystery")) { ... }
        if (category.equals("Romance")) { ... }
        if (category.equals("Science Fiction")) { ... }
        if (category.equals("Young Adult")) { ... }
        if (category.equals("Ghost")) { ... }

        if (category.equals("History")) { ... }
    }
}

```



```

if (category.equals("Philosophy")) {    ...}
if (category.equals("Poetry")) {    ...}
if (category.equals("Epic")) {    ...}
if (category.equals("Adventure")) {    ...}
textView = findViewById(R.id.back_to_categories);
textView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(BooksList.this, Categories.class));    });}}

```

## Upload Book

### Activity\_upload.xml

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".UploadFile">    <Button
        android:id="@+id/select_file"
        android:layout_width="266dp"
        android:layout_height="58dp"
        android:layout_marginStart="72dp"
        android:layout_marginTop="192dp"/>
    <TextView
        android:id="@+id/notify"
        android:layout_width="154dp"
        android:layout_height="78dp"
        android:layout_width="221dp"
        android:layout_height="51dp"
        android:background="@drawable/background_color"
        android:text="Upload"
        android:textColor="@color/white" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

### Uploadbook.java

```

package com.android.goeasy;
import ...;
public class UploadFile extends AppCompatActivity {
    Button selectfile, upload;
    TextView notify;
    Uri pdfUri;
    FirebaseStorage storage;
    FirebaseDatabase database;

```

```

ProgressDialog progressDialog;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_upload_file);
    selectfile = findViewById(R.id.select_file);
    storage = FirebaseStorage.getInstance();
    database = FirebaseDatabase.getInstance();
    upload = findViewById(R.id.upload);
    notify = findViewById(R.id.notify);
    selectfile.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (ContextCompat.checkSelfPermission(UploadFile.this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
                selecctPdf(); } else {
                ActivityCompat.requestPermissions(UploadFile.this, new String[] {
                    Manifest.permission.READ_EXTERNAL_STORAGE
                }, 9);}}});
    upload.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (pdfUri != null) {
                uploadFile(pdfUri);
            } else {
                Toast.makeText(UploadFile.this, "Select a file", Toast.LENGTH_SHORT).show();}}});
private void uploadFile(Uri pdfUri) {
    progressDialog = new ProgressDialog(this);
    progressDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
    progressDialog.setTitle("Uploading File");
    progressDialog.setProgress(0);
    progressDialog.show();
    final String fileName = System.currentTimeMillis() + "";
    StorageReference storageReference = storage.getReference();
    storageReference.child("Uploads").child(fileName).putFile(pdfUri)
        .addOnSuccessListener(new OnSuccessListener < UploadTask.TaskSnapshot > () {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                Task < Uri > url = taskSnapshot.getStorage().getDownloadUrl();
                DatabaseReference reference = database.getReference();
                reference.child(fileName).setValue(url).addOnCompleteListener(new OnCompleteListener <
Void > () {
                    @Override
                    public void onComplete(@NonNull Task < Void > task) {
                        if (task.isSuccessful()) {

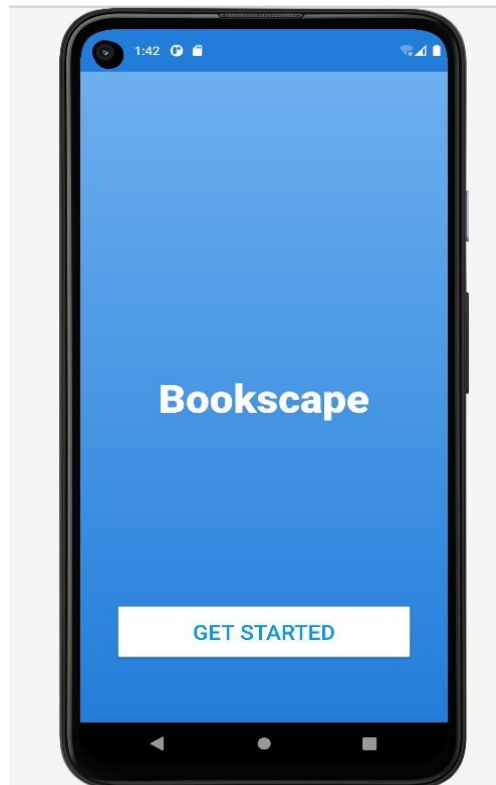
```

```

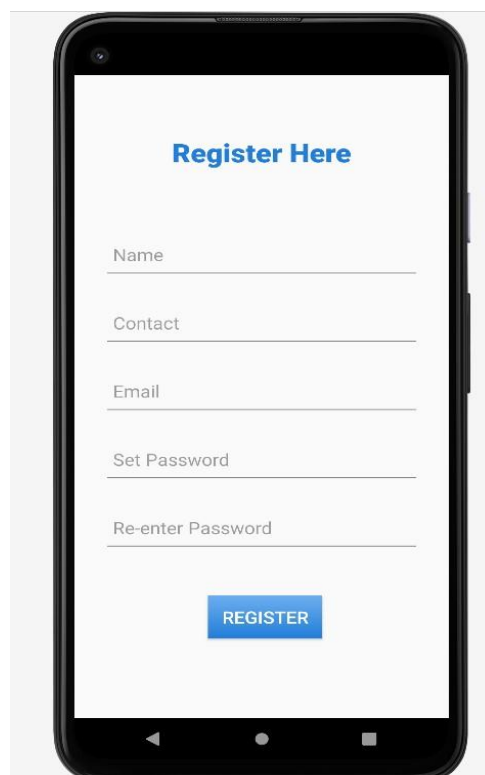
        Toast.makeText(UploadFile.this, "File Successfully uploaded",
Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(UploadFile.this, "File not uploaded",
Toast.LENGTH_SHORT).show();}}});
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText(UploadFile.this, "File not uploaded", Toast.LENGTH_SHORT).show();}
    }).addOnProgressListener(new OnProgressListener < UploadTask.TaskSnapshot > () {
        @Override
        public void onProgress(@NonNull UploadTask.TaskSnapshot snapshot) {
            int currentProgress = (int)(100 * snapshot.getBytesTransferred() /
snapshot.getTotalByteCount());
            progressDialog.setProgress(currentProgress) } }));
    @SuppressWarnings("MissingSuperCall")
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        if (requestCode == 9 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            selecctPdf();
        } else {
            Toast.makeText(UploadFile.this, "Please provide permission",
Toast.LENGTH_SHORT).show();}
        private void selecctPdf() {
            Intent intent = new Intent();
            intent.setType("application/pdf");
            intent.setAction(Intent.ACTION_GET_CONTENT);
            startActivityForResult(intent, 86);}
        @SuppressWarnings("MissingSuperCall")
        @Override
        protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
            if (requestCode == 86 && resultCode == RESULT_OK && data != null) {
                pdfUri = data.getData();
                notify.setText("A file is selected" + data.getData().getLastPathSegment());
            } else {
                Toast.makeText(UploadFile.this, "Please select a file", Toast.LENGTH_SHORT).show();}}}

```

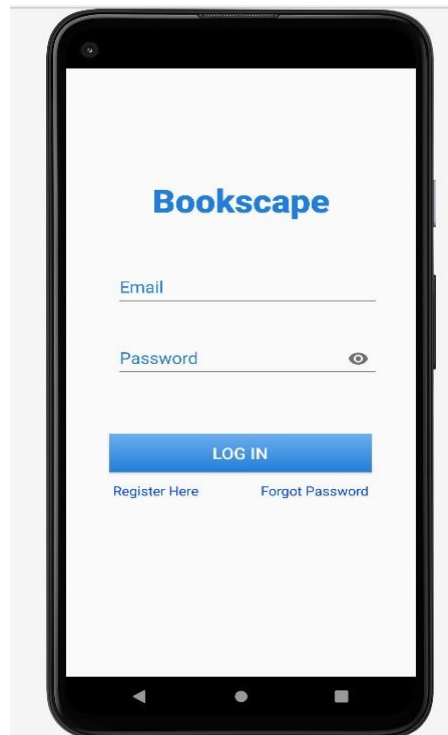
**RESULT :**



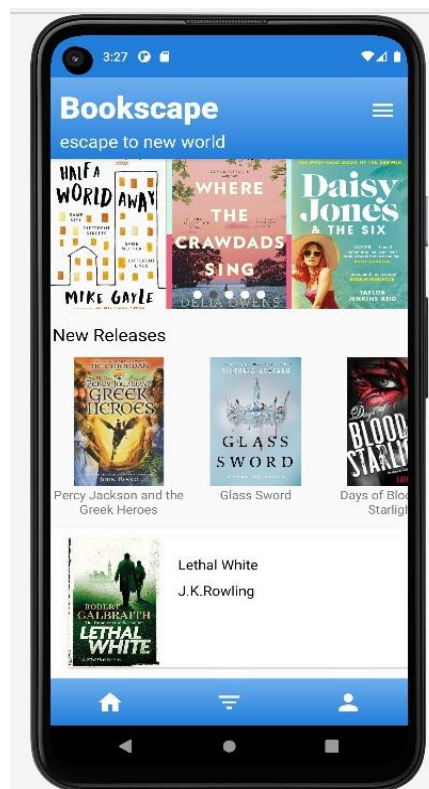
**1.REGISTER**



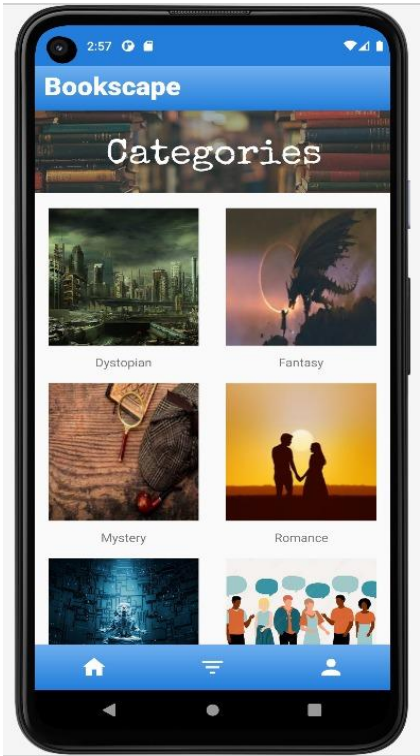
## 2.LOGIN



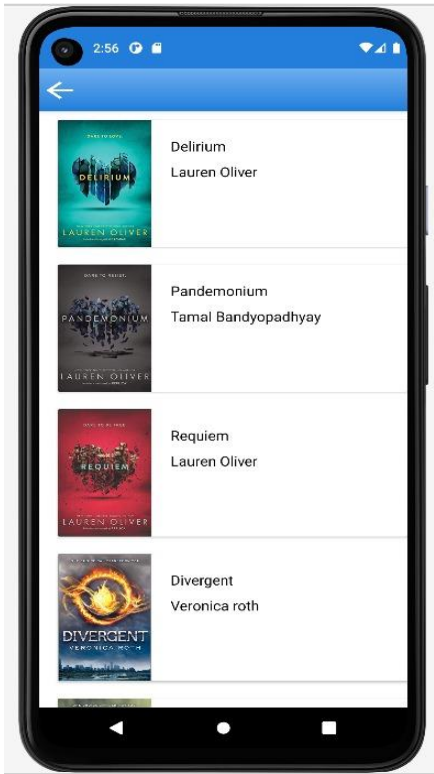
## 3.HOME



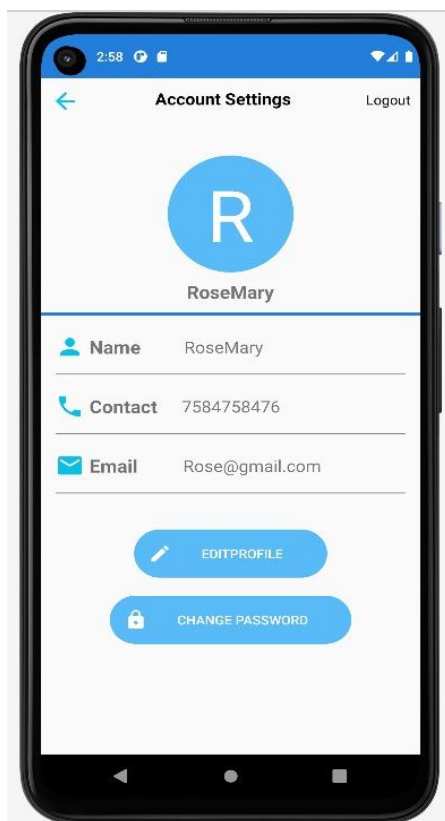
4.CATEGORIES



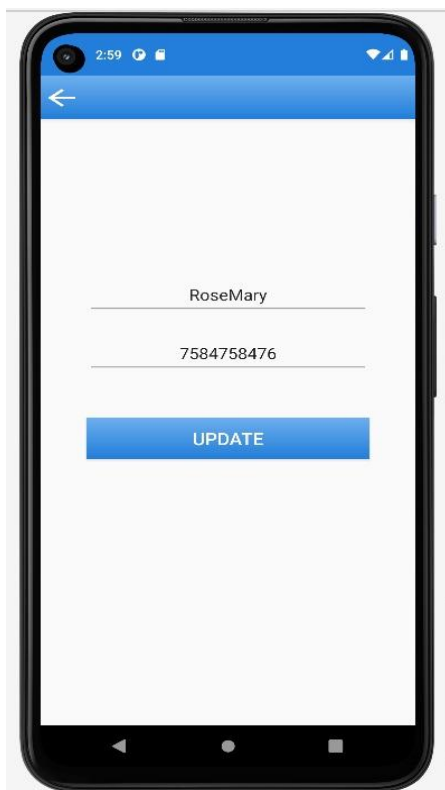
5.LIST OF BOOKS



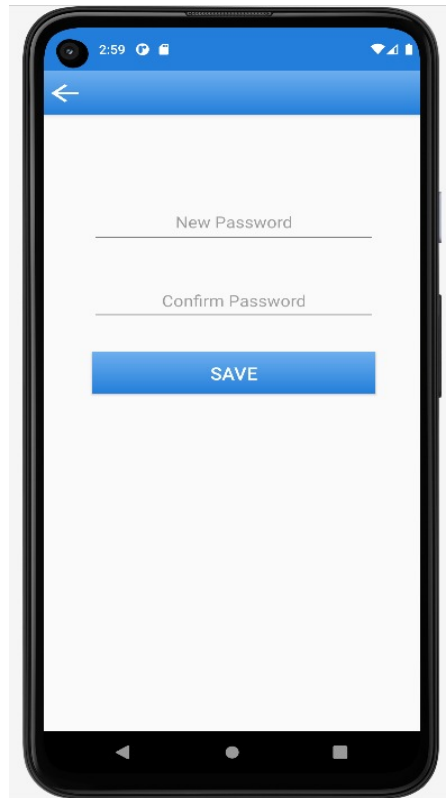
## 6.PROFILE



## 7.EDIT PROFILE

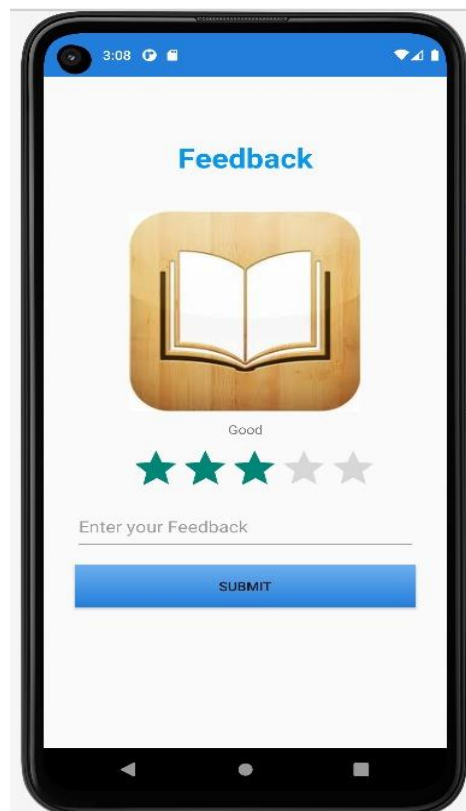


## 8.CHANGE PASSWORD



A mobile app screenshot of the 'CHANGE PASSWORD' screen. The screen has a white background with a blue header bar at the top containing a back arrow and the time 2:59. Below the header, there are two text input fields: 'New Password' and 'Confirm Password'. A blue 'SAVE' button is positioned below the 'Confirm Password' field. The bottom of the screen shows the Android navigation bar.

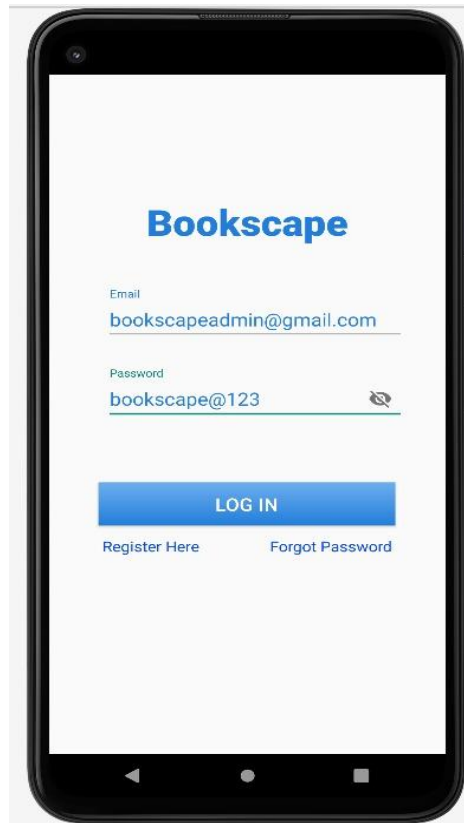
## 9.FEEDBACK



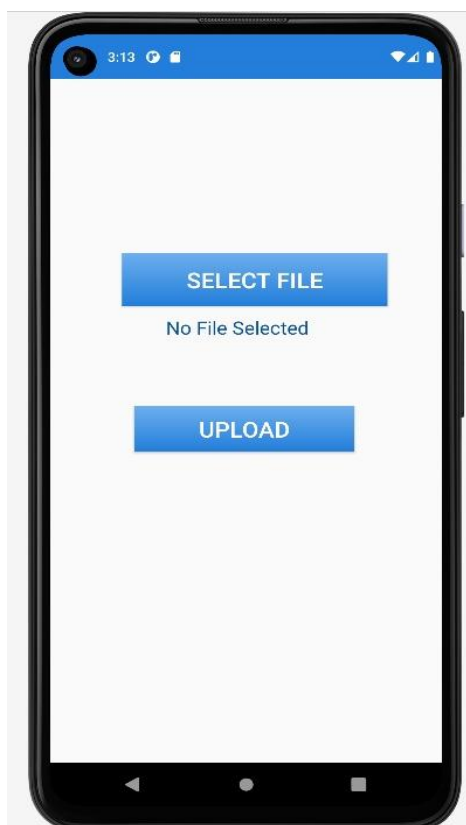
A mobile app screenshot of the 'FEEDBACK' screen. The screen has a white background with a blue header bar at the top containing the time 3:08. Below the header, the word 'Feedback' is displayed in blue. Underneath is a large icon of an open book on a wooden surface. Below the icon, the word 'Good' is centered. There are five stars; the first three are green and the last two are grey. Below the stars is a text input field labeled 'Enter your Feedback'. A blue 'SUBMIT' button is at the bottom. The bottom of the screen shows the Android navigation bar.



## 10.ADMIN LOGIN



## 11.UPLOAD BOOK



## REFERENCES

<https://developer.android.com/>

<https://www.udemy.com/course/full-android-11-masterclass-course-with-java-53-hours/>

<https://www.geeksforgeeks.org/android-tutorial/>