Mahatma Education Society's

Pillai College of Arts, Commerce & Science

(Autonomous)

Affiliated to University of Mumbai

NAAC Accredited 'A' grade (3 cycles) Best
College Award by University of Mumbai
ISO 9001:2015 Certified

CERTIFICATE

This is to certify that Mr. **Jeevan Ramesh Jadhav** of **T.Y B.Sc. C.S**. **Semester VI** has completed the Project of the Subject of **Android Programming** during the academic year 2024- 25 under the guidance of **Prof. Dipti Khiste** is the partial requirement for the fulfilment of the curriculum of the Degree of Bachelor of Science in Computer Science, University of Mumbai.

Place:

Date:

Name & Signature Of faculty	Name & Signature of External	Name & Signature of Coordinator

MAHATMA EDUCATION SOCIETY’S

PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE

(Autonomous) NEW PANVEL

PROJECT REPORT ON

“Book Organizer App”

IN PARTIAL FULFILLMENT OF

BACHELOR OF COMPUTER SCIENCE

SEMESTER 2024-2025

PROJECT GUIDE

Prof. Dipti Khiste

SUBMITTED BY: Jeevan Ramesh Jadhav

ROLL NO: 9125

Evaluation sheet for continuous assessment with rubrics

Class: Computer Science

Subject: Android Programming

Topic: Book Organizer App

Name: Jeevan Ramesh Jadhav

Roll Number: 9125 **Class / Division**: TYCS/A

Name of Evaluator: Prof. Dipti Khiste

Please circle appropriate score

Grading Criteria	Fair	Good	Excellent	Total
Introduction/ Description of the Case	**1**	**2**	**3**	**/3**
SWOT Analysis of the company used for case analysis pertaining to the case (strength of CA2 topic: e.g main important feature of CA1, Weakness: limitations of the project, Opportunities: in carrier in the future, Threat: obstacles that can cause failure to project CA2	**3**	**4**	**5**	**/5**
Learnings from the case	**2**	**3**	**4**	**/4**
Delivery/presentation skills	**1**	**2**	**3**	**/3**
Total				**/15**

- **Inform the class the rubric format and the method of evaluation.**

Pawar

Co-ordinator, Shubhangi Pawar

Book Organizer App

Introduction:

The **Manga Book Organizer App** is an Android application designed to help users efficiently manage and organize their manga collections. It provides features for adding, categorizing, searching, and viewing manga books, making it easier for users to keep track of their favorite titles. The app includes a login system for both admins and users, ensuring secure access and personalized book management. With an intuitive and visually appealing interface, users can navigate seamlessly and organize their manga collections with ease.

Developed using **Android Studio**, the app integrates **Firebase** for real-time data storage and retrieval, ensuring efficient book management. It supports **PDF viewing**, allowing users to read manga directly within the app. Additionally, the search functionality helps users quickly find specific titles, and book categories ensure structured organization. With its user-friendly design and smooth performance, the **Manga Book Organizer App** enhances the overall manga reading and collection experience.

Objective:

The primary objective of this project is to design and develop a lightweight, user-friendly, and efficient mobile application for managing Manga Books. It aims to:

- The admin should get registered in the **Admin** database.
- The admin credentials should be checked in the database and get Login to Admin Dashboard.
- The Admin dashboard should display all the Categories and if searched then a particular category should get filtered out.
- The Admin should get to add new Categories.
- The categories can be able to get delete.
- There should be a floating button which will redirect us to the Page where we can add the particular manga book pdf into the database with, its name, description, category and pdf.
- When ever we will click a particular category then it should show us the manga books details according to the category which we have selected.
- We should be able to logout from the Admin Dashboard when will will click on "Logout button".

Technologies Used:

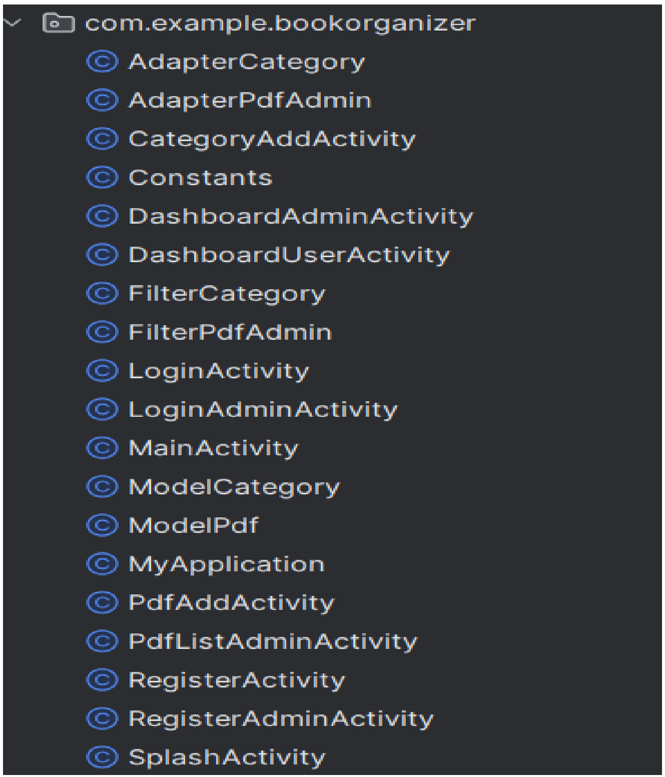
Programming Language: Java (Android)

Database: Firebase Database for RealTime Database storage.

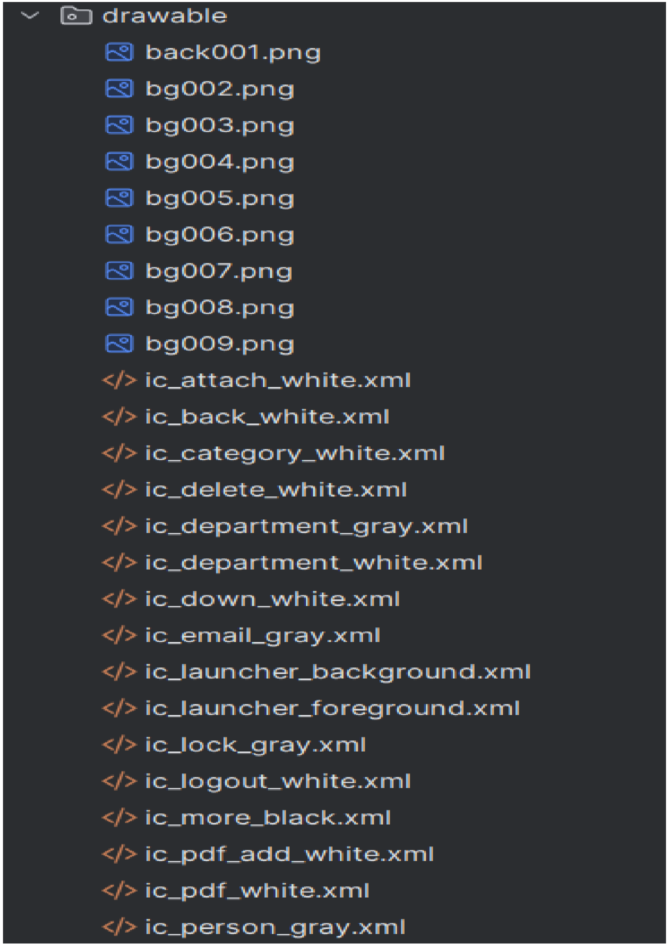
UI Components: LinearLayout, RelativeLayout, ScrollView, CardView, RecyclerView, TextView, ImageView, ImageButton, EditText, TextInputLayout, Button, FloatingActionButton, ProgressBar, DonutProgress, PDFView, and LinearLayoutManager.

Development Tool: Android Studio

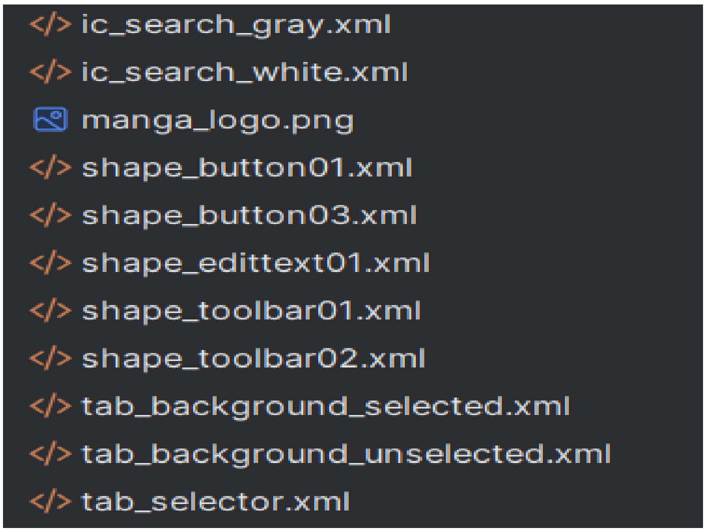
Code:



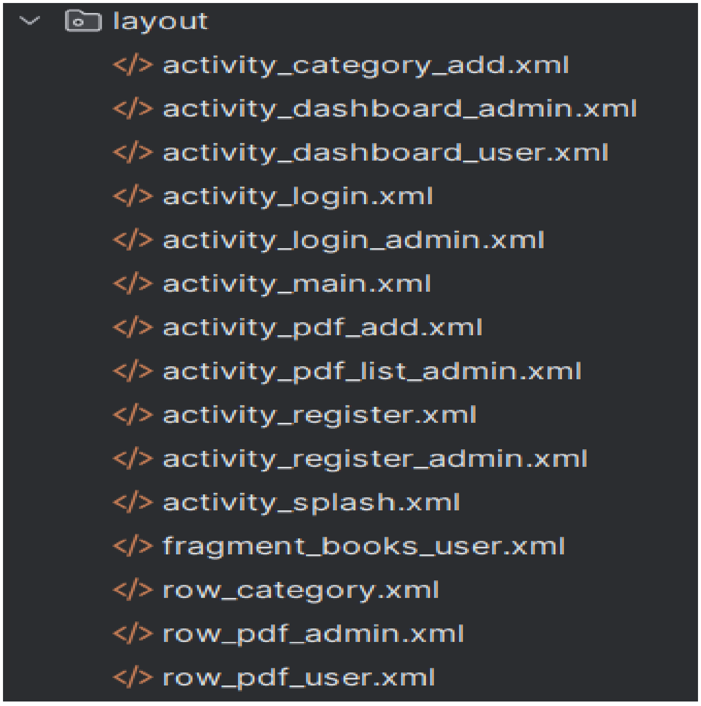
```
com.example.bookorganizer
    AdapterCategory
    AdapterPdfAdmin
    CategoryAddActivity
    Constants
    DashboardAdminActivity
    DashboardUserActivity
    FilterCategory
    FilterPdfAdmin
    LoginActivity
    LoginAdminActivity
    MainActivity
    ModelCategory
    ModelPdf
    MyApplication
    PdfAddActivity
    PdfListAdminActivity
    RegisterActivity
    RegisterAdminActivity
    SplashActivity
```



```
drawable
    back001.png
    bg002.png
    bg003.png
    bg004.png
    bg005.png
    bg006.png
    bg007.png
    bg008.png
    bg009.png
    ic_attach_white.xml
    ic_back_white.xml
    ic_category_white.xml
    ic_delete_white.xml
    ic_department_gray.xml
    ic_department_white.xml
    ic_down_white.xml
    ic_email_gray.xml
    ic_launcher_background.xml
    ic_launcher_foreground.xml
    ic_lock_gray.xml
    ic_logout_white.xml
    ic_more_black.xml
    ic_pdf_add_white.xml
    ic_pdf_white.xml
    ic_person_gray.xml
```



```
ic_search_gray.xml
ic_search_white.xml
manga_logo.png
shape_button01.xml
shape_button03.xml
shape_edittext01.xml
shape_toolbar01.xml
shape_toolbar02.xml
tab_background_selected.xml
tab_background_unselected.xml
tab_selector.xml
```



```
layout
    activity_category_add.xml
    activity_dashboard_admin.xml
    activity_dashboard_user.xml
    activity_login.xml
    activity_login_admin.xml
    activity_main.xml
    activity_pdf_add.xml
    activity_pdf_list_admin.xml
    activity_register.xml
    activity_register_admin.xml
    activity_splash.xml
    fragment_books_user.xml
    row_category.xml
    row_pdf_admin.xml
    row_pdf_user.xml
```

Java Codes:

SplashActivity.java:

```
package com.example.bookorganizer;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class SplashActivity extends AppCompatActivity {

    private FirebaseAuth firebaseAuth;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);

        firebaseAuth = FirebaseAuth.getInstance();

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                checkUser();
            }
        }, 1000);
    }

    private void checkUser() {
        FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();
        if (firebaseUser == null){
            startActivity(new Intent(SplashActivity.this, MainActivity.class));
            finish();
        }
        else{
            DatabaseReference ref = FirebaseDatabase.getInstance("https://book-organizer-
dd855-default-rtdb.asia-southeast1.firebasedatabase.app/")
                    .getReference("Users");
            ref.child(firebaseUser.getUid())
                    .addListenerForSingleValueEvent(new ValueEventListener() {
                        @Override
                        public void onDataChange(@NonNull DataSnapshot snapshot) {

                            String userType = ""+snapshot.child("userType").getValue();

                            if(userType.equals("user")){
                                startActivity(new Intent(SplashActivity.this,
DashboardUserActivity.class));
                                finish();
                            } else if (userType.equals("admin")) {
                                startActivity(new Intent(SplashActivity.this,
DashboardAdminActivity.class));
```

```
                                finish();
                            }
                        }

                        @Override
                        public void onCancelled(@NonNull DatabaseError error) {

                        }
                    });
        }
    }
}
```

MainActivity.java:

```
package com.example.bookorganizer;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import androidx.appcompat.app.AppCompatActivity;

import com.example.bookorganizer.databinding.ActivityMainBinding;

public class MainActivity extends AppCompatActivity {

    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        binding.loginusrbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MainActivity.this, LoginActivity.class));
            }
        });

        binding.loginadminbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MainActivity.this, LoginAdminActivity.class));
            }
        });

        binding.skipbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(MainActivity.this, DashboardUserActivity.class));
            }
        });
    }
}
```

LoginAdminActivity.java:

```
package com.example.bookorganizer;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Patterns;
import android.view.View;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.example.bookorganizer.databinding.ActivityLoginBinding;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class LoginAdminActivity extends AppCompatActivity {

    private ActivityLoginBinding binding;
    private FirebaseAuth firebaseAuth;
    private ProgressDialog progressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityLoginBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        firebaseAuth = FirebaseAuth.getInstance();

        progressDialog = new ProgressDialog(this);
        progressDialog.setTitle("Please Wait");
        progressDialog.setCanceledOnTouchOutside(false);
        // Disable "Create Account" for Admin login
        binding.noAccountTv.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(LoginAdminActivity.this, RegisterAdminActivity.class));
            }
        });

        binding.loginbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                validateData();
            }
        });
    }

    private String email = "", password = "";

    private void validateData() {
        email = binding.emailEt.getText().toString().trim();
        password = binding.passwordEt.getText().toString().trim();

        if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            Toast.makeText(this, "Invalid Email pattern!", Toast.LENGTH_SHORT).show();
        } else if (TextUtils.isEmpty(password)) {
            Toast.makeText(this, "Enter password!", Toast.LENGTH_SHORT).show();
        } else {
            loginUser();
        }
    }

    private void loginUser() {
        progressDialog.setMessage("Logging In...");
        progressDialog.show();

        firebaseAuth.signInWithEmailAndPassword(email, password)
                .addOnSuccessListener(new OnSuccessListener<AuthResult>() {
                    @Override
                    public void onSuccess(AuthResult authResult) {
                        checkAdmin();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        progressDialog.dismiss();
                        Toast.makeText(LoginAdminActivity.this, "" + e.getMessage(),
Toast.LENGTH_SHORT).show();
                    }
            });
    }

    private void checkAdmin() {
        progressDialog.setMessage("Checking Admin...");

        FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();
        if (firebaseUser == null) {
            progressDialog.dismiss();
            Toast.makeText(this, "Authentication Failed!", Toast.LENGTH_SHORT).show();
            return;
        }

        DatabaseReference ref = FirebaseDatabase.getInstance("https://book-organizer-dd855-
default-rtdb.asia-southeast1.firebasedatabase.app/")
                .getReference("Admins");

        ref.child(firebaseUser.getUid()).addListenerForSingleValueEvent(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                progressDialog.dismiss();

                if (snapshot.exists()) {
                    String userType = "" + snapshot.child("userType").getValue();
                    if (userType.equals("admin")) {
                        startActivity(new Intent(LoginAdminActivity.this,
DashboardAdminActivity.class));
                        finish();
                    } else {
                        firebaseAuth.signOut();
                        Toast.makeText(LoginAdminActivity.this, "Access Denied! Not an Admin.",
Toast.LENGTH_SHORT).show();
                    }
                } else {
                    firebaseAuth.signOut();
                    Toast.makeText(LoginAdminActivity.this, "Admin Not Found!",
Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                progressDialog.dismiss();
                Toast.makeText(LoginAdminActivity.this, "Database Error: " + error.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

RegisterAdminActivity.java:

```
package com.example.bookorganizer;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.util.Patterns;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.example.bookorganizer.databinding.ActivityRegisterBinding;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.HashMap;

public class RegisterAdminActivity extends AppCompatActivity {

    // View binding
    private ActivityRegisterBinding binding;

    // Firebase auth
    private FirebaseAuth firebaseAuth;

    private ProgressDialog progressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // ✅ Initialize View Binding before using it
        binding = ActivityRegisterBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        firebaseAuth = FirebaseAuth.getInstance();

        progressDialog = new ProgressDialog(this);
        progressDialog.setTitle("Please Wait");
        progressDialog.setCanceledOnTouchOutside(false);

        binding.backBtn.setOnClickListener(v ->
getOnBackPressedDispatcher().onBackPressed());
```

```
        binding.registerbtn.setOnClickListener(v -> validateData());
    }

    private String name = "", email = "", password = "";

    private void validateData() {
        name = binding.nameEt.getText().toString().trim();
        email = binding.emailEt.getText().toString().trim();
        password = binding.passwordEt.getText().toString().trim();
        String cpassword = binding.cpasswordEt.getText().toString().trim();

        // Validate Data
        if (TextUtils.isEmpty(name)) {
            Toast.makeText(this, "Enter your Name", Toast.LENGTH_SHORT).show();
        } else if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            Toast.makeText(this, "Invalid Email pattern!", Toast.LENGTH_SHORT).show();
        } else if (TextUtils.isEmpty(password)) {
            Toast.makeText(this, "Enter password!", Toast.LENGTH_SHORT).show();
        } else if (TextUtils.isEmpty(cpassword)) {
            Toast.makeText(this, "Confirm Password!", Toast.LENGTH_SHORT).show();
        } else if (!password.equals(cpassword)) {
            Toast.makeText(this, "Passwords don't match!", Toast.LENGTH_SHORT).show();
        } else {
            createAdminAccount();
        }
    }

    private void createAdminAccount() {
        progressDialog.setMessage("Creating Admin Account...");
        progressDialog.show();

        firebaseAuth.createUserWithEmailAndPassword(email, password)
                .addOnSuccessListener(new OnSuccessListener<AuthResult>() {
                    @Override
                    public void onSuccess(AuthResult authResult) {
                        Log.d("FirebaseAuth", "Admin registered successfully: " +
authResult.getUser().getUid());
                        saveAdminInfo();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        progressDialog.dismiss();
                        Log.e("FirebaseAuth", "Registration failed: " + e.getMessage());
                        Toast.makeText(RegisterAdminActivity.this, "" + e.getMessage(),
Toast.LENGTH_SHORT).show();
                    }
                });
    }

    private void saveAdminInfo() {
        progressDialog.setMessage("Saving Admin Info...");

        long timestamp = System.currentTimeMillis();

        // ✅ Ensure user is authenticated before getting UID
        if (firebaseAuth.getCurrentUser() == null) {
            progressDialog.dismiss();
            Toast.makeText(this, "Error: User not authenticated!",
Toast.LENGTH_SHORT).show();
            return;
        }

        String uid = firebaseAuth.getCurrentUser().getUid();

        HashMap<String, Object> hashMap = new HashMap<>();
        hashMap.put("uid", uid);
        hashMap.put("email", email);
        hashMap.put("name", name);
        hashMap.put("profileImage", ""); // Can add default image URL
        hashMap.put("userType", "admin"); // ✅ Changed from "user" to "admin"
        hashMap.put("timestamp", timestamp);

        // ✅ Storing Admin data in "Admins" node instead of "Users"
        DatabaseReference ref = FirebaseDatabase.getInstance("https://book-organizer-dd855-
default-rtdb.asia-southeast1.firebasedatabase.app/")
                .getReference("Admins");
        ref.child(uid)
                .setValue(hashMap)
                .addOnSuccessListener(unused -> {
                    progressDialog.dismiss();
                    Toast.makeText(RegisterAdminActivity.this, "Admin Account Created!",
Toast.LENGTH_SHORT).show();
                    startActivity(new Intent(RegisterAdminActivity.this,
LoginAdminActivity.class)); // ✅ Redirect to Admin Login
                    finish();
                })
                .addOnFailureListener(e -> {
                    progressDialog.dismiss();
                    Toast.makeText(RegisterAdminActivity.this, "" + e.getMessage(),
Toast.LENGTH_SHORT).show();
                });
    }
}
```

DashboardAdminActivity:

```
package com.example.bookorganizer;

import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;

import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import com.example.bookorganizer.databinding.ActivityDashboardAdminBinding;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class DashboardAdminActivity extends AppCompatActivity {

    private ActivityDashboardAdminBinding binding;

    private FirebaseAuth firebaseAuth;

    private ArrayList<ModelCategory> categoryArrayList;

    private AdapterCategory adapterCategory;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityDashboardAdminBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        firebaseAuth = FirebaseAuth.getInstance();
        checkUser();
        loadCategories();

        binding.searchET.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {
            }

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {
                try {
                    adapterCategory.getFilter().filter(s);
                }
                catch (Exception e){

                }
            }

            @Override
            public void afterTextChanged(Editable s) {

            }
        });

        binding.logoutBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                firebaseAuth.signOut();
                checkUser();
            }
        });

        binding.addCategoryBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(DashboardAdminActivity.this,
CategoryAddActivity.class));
            }
        });

        binding.addPdfFab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                startActivity(new Intent(DashboardAdminActivity.this, PdfAddActivity.class));
            }
        });

    }

    private void loadCategories() {
        categoryArrayList = new ArrayList<>();

        DatabaseReference ref = FirebaseDatabase.getInstance("https://book-organizer-dd855-
default-rtdb.asia-southeast1.firebasedatabase.app/")
                .getReference("Categories");
        ref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                categoryArrayList.clear();
                for (DataSnapshot ds: snapshot.getChildren()){
                    ModelCategory model = ds.getValue(ModelCategory.class);

                    categoryArrayList.add(model);
                }

                adapterCategory = new AdapterCategory(DashboardAdminActivity.this,
categoryArrayList);
                binding.categoriesRv.setAdapter(adapterCategory);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
    }

    private void checkUser() {
        FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();
        if (firebaseUser == null){
            startActivity(new Intent(DashboardAdminActivity.this, MainActivity.class));
            finish();
        }
        else {
            String email = firebaseUser.getEmail();

            binding.subTitleTv.setText(email);
        }
    }
}
```

FilterCategory.java:

```
package com.example.bookorganizer;

import android.annotation.SuppressLint;
import android.widget.Filter;

import java.util.ArrayList;

public class FilterCategory extends Filter {
    ArrayList<ModelCategory> filterlist;

    AdapterCategory adapterCategory;

    public FilterCategory(ArrayList<ModelCategory> filterlist, AdapterCategory
adapterCategory) {
        this.filterlist = filterlist;
        this.adapterCategory = adapterCategory;
    }

    @Override
    protected FilterResults performFiltering(CharSequence constraint) {
        FilterResults results = new FilterResults();
        if (constraint != null && constraint.length() > 0){
            constraint = constraint.toString().toUpperCase();
            ArrayList<ModelCategory> filteredModels = new ArrayList<>();

            for (int i=0; i<filterlist.size(); i++){
                if (filterlist.get(i).getCategory().toUpperCase().contains(constraint)){
                    filteredModels.add(filterlist.get(i));
                }
            }
            results.count = filteredModels.size();
            results.values = filteredModels;
         }
        else {
            results.count = filterlist.size();
            results.values = filterlist;
        }
        return results;
    }

    @SuppressLint("NotifyDataSetChanged")
    @Override
    protected void publishResults(CharSequence constraint, FilterResults results) {
        adapterCategory.categoryArrayList = (ArrayList<ModelCategory>)results.values;

        adapterCategory.notifyDataSetChanged();
    }
}
```

AdapterCategory.java:

```
package com.example.bookorganizer;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;


import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.bookorganizer.databinding.RowCategoryBinding;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.ArrayList;

public class AdapterCategory extends
RecyclerView.Adapter<AdapterCategory.HolderCategory> implements Filterable {

    private Context context;
    public ArrayList<ModelCategory> categoryArrayList, filterlist;

    private RowCategoryBinding binding;

    private FilterCategory filter;

    public AdapterCategory(Context context, ArrayList<ModelCategory> categoryArrayList)
{
        this.context = context;
        this.categoryArrayList = categoryArrayList;
        this.filterlist = categoryArrayList;
    }

    @NonNull
    @Override
    public HolderCategory onCreateViewHolder(@NonNull ViewGroup parent, int viewType)
{
        binding = RowCategoryBinding.inflate(LayoutInflater.from(context), parent, false);
        return new HolderCategory(binding.getRoot());
    }

    @Override
    public void onBindViewHolder(@NonNull HolderCategory holder, int position) {
        ModelCategory model = categoryArrayList.get(position);
        String id = model.getId();
        String category = model.getCategory();
        String uid = model.getUid();
        long timestamp = model.getTimestamp();

        holder.categoryTv.setText(category);

        holder.deleteBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AlertDialog.Builder builder = new AlertDialog.Builder(context);
                builder.setTitle("Delete")
                        .setMessage("Are you sure you want to delete this Category?")
                        .setPositiveButton("Confirm", new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int which) {
                                Toast.makeText(context, "Deleting...", Toast.LENGTH_SHORT).show();
                                deleteCategory(model, holder);
                            }
                        })
                        .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialog, int which) {
                                dialog.dismiss();
                            }
                        })
                        .show();
            }
        });
        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(context, PdfListAdminActivity.class);
                intent.putExtra("categoryId", id);
                intent.putExtra("categoryTitle", category);
                context.startActivity(intent);
            }
        });
    }

    private void deleteCategory(ModelCategory model, HolderCategory holder) {
        String id = model.getId();

        DatabaseReference ref = FirebaseDatabase.getInstance("https://book-organizer-dd855-
default-rtdb.asia-southeast1.firebasedatabase.app/")
                .getReference("Categories");
        ref.child(id)
                .removeValue()
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void unused) {
                        Toast.makeText(context, "Successfully Deleted..",
Toast.LENGTH_SHORT).show();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        Toast.makeText(context, ""+e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                });
    }

    @Override
    public int getItemCount() {
        return categoryArrayList.size();
    }

    @Override
    public Filter getFilter() {
        if (filter == null){
            filter = new FilterCategory(filterlist, this);
        }
        return filter;
    }

    class HolderCategory extends RecyclerView.ViewHolder {

        TextView categoryTv;
        ImageButton deleteBtn;

        public HolderCategory(@NonNull View itemView) {
            super(itemView);

            categoryTv = binding.categoryTv;
            deleteBtn = binding.deleteBtn;
        }
    }
}
```

PdfAddActivity.java:

```
package com.example.bookorganizer;

import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import com.example.bookorganizer.databinding.ActivityPdfAddBinding;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.util.ArrayList;
import java.util.HashMap;

public class PdfAddActivity extends AppCompatActivity {

    private ActivityPdfAddBinding binding;
    private FirebaseAuth firebaseAuth;
    private ProgressDialog progressDialog;
    private ArrayList<String> categoryTitleArrayList, categoryIdArrayList;
    private Uri pdfUri = null;
    private static final int PDF_PICK_CODE = 1000;
    private static final String TAG = "ADD_PDF_TAG";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityPdfAddBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        firebaseAuth = FirebaseAuth.getInstance();
        loadPdfCategories();

        progressDialog = new ProgressDialog(this);
        progressDialog.setTitle("Please Wait");
        progressDialog.setCanceledOnTouchOutside(false);

        binding.backBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                onBackPressed();
            }
        });

        binding.attachBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
            pdfPickIntent();
        }
    });

    binding.categoryTv.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            categoryPickDialog();
        }
    });

    binding.submitBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            validateData();
        }
    });
}

private String title = "", description = "";

private void validateData() {
    Log.d(TAG, "validateData: Validating Data....");

    title = binding.titleEt.getText().toString().trim();
    description = binding.descriptionEt.getText().toString().trim();

    if (TextUtils.isEmpty(title)) {
        Toast.makeText(this, "Enter Title....", Toast.LENGTH_SHORT).show();
    } else if (TextUtils.isEmpty(description)) {
        Toast.makeText(this, "Enter Description...", Toast.LENGTH_SHORT).show();
    } else if (TextUtils.isEmpty(selectedCategoryTitle)) {
        Toast.makeText(this, "Pick Category....", Toast.LENGTH_SHORT).show();
    } else if (pdfUri == null) {
        Toast.makeText(this, "Pick PDF....", Toast.LENGTH_SHORT).show();
    } else {
        uploadPdfToStorage();
    }
}

private void uploadPdfToStorage() {
    Log.d(TAG, "uploadPdfToStorage: Uploading to Storage....");

    progressDialog.setMessage("Uploading PDF....");
    progressDialog.show();

    long timestamp = System.currentTimeMillis();

    String filePathAndName = "Books/" + timestamp;
```

```
        // Use the default instance of FirebaseStorage
        StorageReference storageReference = FirebaseStorage.getInstance("gs://book-organizer-
dd855.firebasestorage.app/").getReference(filePathAndName);
        storageReference.putFile(pdfUri)
                .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
                    @Override
                    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
                        Log.d(TAG, "onSuccess: PDF uploaded to Storage..");
                        Log.d(TAG, "onSuccess: getting PDF url");

                        Task<Uri> uriTask = taskSnapshot.getStorage().getDownloadUrl();
                        while(!uriTask.isSuccessful());
                        String uploadedPdfUrl = ""+uriTask.getResult();

                        uploadPdfInfoToDb(uploadedPdfUrl, timestamp);
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        progressDialog.dismiss();
                        Log.e(TAG, "onFailure: PDF upload Failed due to " + e.getMessage());
                        Toast.makeText(PdfAddActivity.this, "PDF upload Failed due to " +
e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                });
    }

    private void uploadPdfInfoToDb(String uploadedPdfUrl, long timestamp) {
        Log.d(TAG, "uploadPdfInfoToDb: Uploading PDF Info to firebase db....");

        progressDialog.setMessage("Uploading PDF info...");

        String uid = firebaseAuth.getUid();

        HashMap<String, Object>hashMap = new HashMap<>();
        hashMap.put("uid", ""+uid);
        hashMap.put("id", ""+timestamp);
        hashMap.put("title", ""+title);
        hashMap.put("description", ""+description);
        hashMap.put("categoryId", ""+selectedCategoryId);
        hashMap.put("url", ""+uploadedPdfUrl);
        hashMap.put("timestamp", timestamp);

        DatabaseReference ref = FirebaseDatabase.getInstance("https://book-organizer-dd855-
default-rtdb.asia-southeast1.firebasedatabase.app/")
                .getReference("Books");
        ref.child(""+timestamp)
                .setValue(hashMap)
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void unused) {
                        progressDialog.dismiss();
                        Log.d(TAG, "onSuccess: Successfully uploaded...");
                        Toast.makeText(PdfAddActivity.this, "Successfully uploaded...",
Toast.LENGTH_SHORT).show();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        progressDialog.dismiss();
                        Log.d(TAG, "onFailure: Failed to upload to db due to "+e.getMessage());
                        Toast.makeText(PdfAddActivity.this, "Failed to upload to db due to
"+e.getMessage(), Toast.LENGTH_SHORT).show();
                    }
                });
    }


    private void loadPdfCategories() {
        Log.d(TAG, "loadPdfCategories: Loading PDF Categories...");
        categoryTitleArrayList = new ArrayList<>();
        categoryIdArrayList = new ArrayList<>();

        DatabaseReference ref = FirebaseDatabase.getInstance("https://book-organizer-dd855-
default-rtdb.asia-southeast1.firebasedatabase.app/")
                .getReference("Categories");
        ref.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                categoryTitleArrayList.clear();
                categoryIdArrayList.clear();
                for (DataSnapshot ds : snapshot.getChildren()) {

                    String categoryId = ""+ds.child("id").getValue();
                    String categoryTitle = ""+ds.child("category").getValue();

                    categoryTitleArrayList.add(categoryTitle);
                    categoryIdArrayList.add(categoryId);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Log.e(TAG, "onCancelled: " + error.getMessage());
            }
        });
    }
    private String selectedCategoryId, selectedCategoryTitle;
    private void categoryPickDialog() {
        Log.d(TAG, "categoryPickDialog: Showing Category Pick Dialog");

        String[] categoriesArray = new String[categoryTitleArrayList.size()];
        for (int i = 0; i < categoryTitleArrayList.size(); i++) {
            categoriesArray[i] = categoryTitleArrayList.get(i);
        }

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Pick Category")
                .setItems(categoriesArray, new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        selectedCategoryTitle = categoryTitleArrayList.get(which);
                        selectedCategoryId = categoryIdArrayList.get(which);
                        binding.categoryTv.setText(selectedCategoryTitle);
                        Log.d(TAG, "onClick: Selected Category: " +selectedCategoryId+"
"+selectedCategoryTitle);
                    }
                })
                .show();
    }

    private void pdfPickIntent() {
        Log.d(TAG, "pdfPickIntent: starting pdf pick intent");

        Intent intent = new Intent();
        intent.setType("application/pdf");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Select Pdf"), PDF_PICK_CODE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (resultCode == RESULT_OK) {
            if (requestCode == PDF_PICK_CODE) {
                Log.d(TAG, "onActivityResult: PDF picked");
                pdfUri = data.getData();
                Log.d(TAG, "onActivityResult: URI: " + pdfUri);
            }
        }
        else {
            Log.d(TAG, "onActivityResult: Cancelled Picking PDF");
            Toast.makeText(this, "Cancelled Picking PDF", Toast.LENGTH_SHORT).show();
        }
    }
}
```

PdfListAdminActivity.java:

```
package com.example.bookorganizer;

import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.util.Log;
import android.view.View;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.example.bookorganizer.databinding.ActivityPdfListAdminBinding;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class PdfListAdminActivity extends AppCompatActivity {
    private ActivityPdfListAdminBinding binding;

    private ArrayList<ModelPdf> pdfArrayList;
    private AdapterPdfAdmin adapterPdfAdmin;
    private String categoryId, categoryTitle;

    private static final String TAG = "PDF_LIST_TAG";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityPdfListAdminBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        Intent intent = getIntent();
        categoryId = intent.getStringExtra("categoryId");
        categoryTitle = intent.getStringExtra("categoryTitle");

        if (categoryId == null || categoryId.isEmpty()) {
            Toast.makeText(this, "Error: Category ID is missing!",
Toast.LENGTH_SHORT).show();
            finish(); // Close activity if there's no valid category
            return;
        }

        binding.subtitleTv.setText(categoryTitle);
        // Load PDF list for the category
        loadPdfList();

        // Search functionality
        binding.searchEt.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence s, int start, int count, int after) {}

            @Override
            public void onTextChanged(CharSequence s, int start, int before, int count) {
                if (adapterPdfAdmin != null) {
                    adapterPdfAdmin.getFilter().filter(s);
                }
            }

            @Override
            public void afterTextChanged(Editable s) {}
        });

        // Back Button Click
        binding.backBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                onBackPressed();
            }
        });
    }

    private void loadPdfList() {
        pdfArrayList = new ArrayList<>();

        DatabaseReference ref = FirebaseDatabase.getInstance("https://book-organizer-dd855-
default-rtdb.asia-southeast1.firebasedatabase.app/")
                .getReference("Books");

        ref.orderByChild("categoryId").equalTo(categoryId)
                .addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot snapshot) {
                        pdfArrayList.clear();

                        if (!snapshot.exists()) {
                            Toast.makeText(PdfListAdminActivity.this, "No books found!",
Toast.LENGTH_SHORT).show();
                        }

                        for (DataSnapshot ds : snapshot.getChildren()) {
                            ModelPdf model = ds.getValue(ModelPdf.class);
                            if (model != null) {
                                pdfArrayList.add(model);
                                Log.d(TAG, "Loaded PDF: " + model.getId() + " - " + model.getTitle());
                            }
                        }

                        if (adapterPdfAdmin == null) {
                            adapterPdfAdmin = new AdapterPdfAdmin(PdfListAdminActivity.this,
pdfArrayList);
                            binding.bookRv.setAdapter(adapterPdfAdmin);
                        } else {
                            adapterPdfAdmin.notifyDataSetChanged();
                        }
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError error) {
                        Log.e(TAG, "Firebase Error: " + error.getMessage());
                        Toast.makeText(PdfListAdminActivity.this, "Failed to load books!",
Toast.LENGTH_SHORT).show();
                    }
                });
    }
}
```

AdapterPdfAdmin.java:

```
package com.example.bookorganizer;

import android.content.Context;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageButton;
import android.widget.ProgressBar;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.example.bookorganizer.databinding.RowPdfAdminBinding;
import com.github.barteksc.pdfviewer.PDFView;
import com.github.barteksc.pdfviewer.listener.OnErrorListener;
import com.github.barteksc.pdfviewer.listener.OnLoadCompleteListener;
import com.github.barteksc.pdfviewer.listener.OnPageErrorListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageMetadata;
import com.google.firebase.storage.StorageReference;

import java.util.ArrayList;

public class AdapterPdfAdmin extends
RecyclerView.Adapter<AdapterPdfAdmin.HolderPdfAdmin> implements Filterable {

    public static final long MAX_BYTES_PDF = 50000000;
    private final Context context;
    public ArrayList<ModelPdf> pdfArrayList, filterList;
    private FilterPdfAdmin filter;

    private static final String TAG = "PDF_ADAPTER_TAG";

    public AdapterPdfAdmin(Context context, ArrayList<ModelPdf> pdfArrayList) {
        this.context = context;
        this.pdfArrayList = pdfArrayList;
        this.filterList = pdfArrayList;
    }

    @NonNull
    @Override
    public HolderPdfAdmin onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        RowPdfAdminBinding binding =
RowPdfAdminBinding.inflate(LayoutInflater.from(context), parent, false);
        return new HolderPdfAdmin(binding);
    }

    @Override
    public void onBindViewHolder(@NonNull HolderPdfAdmin holder, int position) {
        ModelPdf model = pdfArrayList.get(position);
        String title = model.getTitle();
        String description = model.getDescription();
        long timestamp = model.getTimestamp();

        String formattedDate = MyApplication.formatTimestamp(timestamp);

        holder.titleTv.setText(title);
        holder.descriptionTv.setText(description);
        holder.dateTv.setText(formattedDate);

        loadCategory(model, holder);
        loadPdfFromUrl(model, holder);
        loadPdfSize(model, holder);
    }

    private void loadPdfSize(ModelPdf model, HolderPdfAdmin holder) {
        String pdfUrl = model.getUrl();

        StorageReference storageReference = FirebaseStorage.getInstance("gs://book-organizer-
dd855.firebasestorage.app")
                .getReferenceFromUrl(pdfUrl);
        storageReference.getMetadata()
                .addOnSuccessListener(new OnSuccessListener<StorageMetadata>() {
                    @Override
                    public void onSuccess(StorageMetadata storageMetadata) {
                        double bytes = storageMetadata.getSizeBytes();
                        double kb = bytes / 1024;
                        double mb = kb / 1024;

                        if (mb >= 1) {
                            holder.sizeTv.setText(String.format("%.2f MB", mb));
                        } else if (kb >= 1) {
                            holder.sizeTv.setText(String.format("%.2f KB", kb));
                        } else {
                            holder.sizeTv.setText(String.format("%.2f bytes", bytes));
                        }
                    }
                })
                .addOnFailureListener(e -> Log.e(TAG, "Failed to get PDF size: " +
e.getMessage()));
    }

    private void loadPdfFromUrl(ModelPdf model, HolderPdfAdmin holder) {
        String pdfUrl = model.getUrl();
        holder.progressBar.setVisibility(View.VISIBLE);

        StorageReference storageReference = FirebaseStorage.getInstance("gs://book-organizer-
dd855.firebasestorage.app").getReferenceFromUrl(pdfUrl);
        storageReference.getBytes(MAX_BYTES_PDF)
                .addOnSuccessListener(bytes -> {
                    holder.pdfView.fromBytes(bytes)
                            .pages(0)
                            .spacing(10)
                            .swipeHorizontal(false)
                            .enableSwipe(true)
                            .onError(t -> {
                                holder.progressBar.setVisibility(View.INVISIBLE);
                                Log.e(TAG, "Error loading PDF: " + t.getMessage());
                            })
                            .onPageError((page, t) -> Log.e(TAG, "Page error: " + t.getMessage()))
                            .onLoad(nbPages -> {
                                holder.progressBar.setVisibility(View.INVISIBLE);
                                Log.d(TAG, "PDF Loaded Successfully");
                            })
                            .load();
                })
                .addOnFailureListener(e -> {
                    holder.progressBar.setVisibility(View.INVISIBLE);
                    Log.e(TAG, "Failed to load PDF: " + e.getMessage());
                });
    }

    private void loadCategory(ModelPdf model, HolderPdfAdmin holder) {
        String categoryId = model.getCategoryId();

        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("Categories");
        ref.child(categoryId).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                String category = snapshot.child("category").getValue(String.class);
                holder.categoryTv.setText(category);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Log.e(TAG, "Failed to load category: " + error.getMessage());
            }
        });
    }

    @Override
    public int getItemCount() {
        return pdfArrayList.size();
    }

    @Override
    public Filter getFilter() {
        if (filter == null) {
            filter = new FilterPdfAdmin(filterList, this);
        }
        return filter;
    }

    static class HolderPdfAdmin extends RecyclerView.ViewHolder {
        PDFView pdfView;
        ProgressBar progressBar;
        TextView titleTv, descriptionTv, categoryTv, sizeTv, dateTv;
        ImageButton moreBtn;

        public HolderPdfAdmin(@NonNull RowPdfAdminBinding binding) {
            super(binding.getRoot());
            pdfView = binding.pdfView;
            progressBar = binding.progressBar;
```

```
            titleTv = binding.titleTv;
            descriptionTv = binding.descriptionTv;
            categoryTv = binding.categoryTv;
            sizeTv = binding.sizeTv;
            dateTv = binding.dateTv;
            moreBtn = binding.moreBtn;
        }
    }
}
```

CategoryAddAmin.java:

```
package com.example.bookorganizer;

import android.app.ProgressDialog;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import com.example.bookorganizer.databinding.ActivityCategoryAddBinding;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.HashMap;

public class CategoryAddActivity extends AppCompatActivity {

    private ActivityCategoryAddBinding binding;

    private FirebaseAuth firebaseAuth;

    private ProgressDialog progressDialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityCategoryAddBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        firebaseAuth = FirebaseAuth.getInstance();
        progressDialog = new ProgressDialog(this);
        progressDialog.setTitle("Please Wait");
        progressDialog.setCanceledOnTouchOutside(false);

        binding.backBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                onBackPressed();
            }
        });

        binding.submitBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                validateData();
            }
        });

    }

    private String category = "";

    private void validateData() {
        category = binding.categoryET.getText().toString().trim();
        if (TextUtils.isEmpty(category)){
            Toast.makeText(this, "Please Enter Category....", Toast.LENGTH_SHORT).show();
        }
        else {
            addCategoryFirebase();
        }
    }

    private void addCategoryFirebase() {
        progressDialog.setMessage("Adding Category....");
        progressDialog.show();

        long timestamp = System.currentTimeMillis();

        HashMap<String, Object> hashMap = new HashMap<>();
        hashMap.put("id", ""+timestamp);
        hashMap.put("category", ""+category);
        hashMap.put("timestamp", timestamp);
        hashMap.put("id", ""+timestamp);
        hashMap.put("uid", ""+firebaseAuth.getUid());

        DatabaseReference ref = FirebaseDatabase.getInstance("https://book-organizer-dd855-
default-rtdb.asia-southeast1.firebasedatabase.app/")
                .getReference("Categories");
        ref.child(""+timestamp)
```

```
                .setValue(hashMap)
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void unused) {
                        progressDialog.dismiss();
                        Toast.makeText(CategoryAddActivity.this, "Category Added Successfully...",
Toast.LENGTH_SHORT).show();
                    }
                })
                .addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {
                        progressDialog.dismiss();
                        Toast.makeText(CategoryAddActivity.this, ""+e.getMessage(),
Toast.LENGTH_SHORT).show();
                    }
                });
    }
}
```

FilterPdfAdmin.java:

```
package com.example.bookorganizer;

import android.annotation.SuppressLint;
import android.widget.Filter;
import java.util.ArrayList;

public class FilterPdfAdmin extends Filter {
    private final ArrayList<ModelPdf> originalList;
    private final AdapterPdfAdmin adapterPdfAdmin;

    public FilterPdfAdmin(ArrayList<ModelPdf> originalList, AdapterPdfAdmin
adapterPdfAdmin) {
        this.originalList = new ArrayList<>(originalList); // Keep original list intact
        this.adapterPdfAdmin = adapterPdfAdmin;
    }

    @Override
    protected FilterResults performFiltering(CharSequence constraint) {
        FilterResults results = new FilterResults();

        if (constraint != null && constraint.length() > 0) {
            String searchQuery = constraint.toString().toLowerCase().trim();
            ArrayList<ModelPdf> filteredModels = new ArrayList<>();

            for (ModelPdf pdf : originalList) {
                if (pdf.getTitle().toLowerCase().contains(searchQuery)) {
                    filteredModels.add(pdf);
                }
            }
            results.count = filteredModels.size();
            results.values = filteredModels;
        } else {
            // If no filter query, return full list
            results.count = originalList.size();
            results.values = originalList;
        }
        return results;
    }

    @SuppressLint("NotifyDataSetChanged")
    @Override
    protected void publishResults(CharSequence constraint, FilterResults results) {
        if (results.values != null) {
            adapterPdfAdmin.pdfArrayList = (ArrayList<ModelPdf>) results.values;
            adapterPdfAdmin.notifyDataSetChanged();
        }
    }
}
```

ModelCategory.java:

```
package com.example.bookorganizer;

public class ModelCategory {
    String id, category, uid;
    long timestamp;

    public ModelCategory() {
    }

    public ModelCategory(String id, String category, String uid, long timestamp) {
        this.id = id;
        this.category = category;
        this.uid = uid;
        this.timestamp = timestamp;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getCategory() {
        return category;
    }
    public void setCategory(String category) {
        this.category = category;
    }

    public String getUid() {
        return uid;
    }

    public void setUid(String uid) {
        this.uid = uid;
    }

    public long getTimestamp() {
        return timestamp;
    }

    public void setTimestamp(long timestamp) {
        this.timestamp = timestamp;
    }
}
```

ModelPdf.java:

```
package com.example.bookorganizer;

public class ModelPdf {

    String uid, id, title, description, categoryId, url;
    long timestamp;

    public ModelPdf() {
    }

    public ModelPdf(String uid, String id, String title, String description, String categoryId,
String url, long timestamp) {
        this.uid = uid;
        this.id = id;
        this.title = title;
        this.description = description;
        this.categoryId = categoryId;
        this.url = url;
        this.timestamp = timestamp;
    }

    public String getUid() {
        return uid;
    }

    public void setUid(String uid) {
        this.uid = uid;
    }
    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getCategoryId() {
        return categoryId;
    }

    public void setCategoryId(String categoryId) {
        this.categoryId = categoryId;
    }

    public String getUrl() {
        return url;
    }

    public void setUrl(String url) {
        this.url = url;
    }

    public long getTimestamp() {
        return timestamp;
    }

    public void setTimestamp(long timestamp) {
        this.timestamp = timestamp;
    }
}
```

MyApplication.java:

```
package com.example.bookorganizer;

import android.app.Application;
import android.text.format.DateFormat;

import java.util.Calendar;
import java.util.Locale;

public class MyApplication extends Application {
   @Override
   public void onCreate() {
      super.onCreate();
   }

   public static final String formatTimestamp(long timestamp){
      Calendar cal = Calendar.getInstance(Locale.ENGLISH);
      cal.setTimeInMillis(timestamp);

      String date = DateFormat.format("dd/MM/yyyy", cal).toString();

      return date;
   }
}
```

Layout:

Activity_splash.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:id="@+id/main"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:gravity="center"
   android:background="@drawable/back001"
   tools:context=".SplashActivity">

   <ImageView
      android:layout_width="200dp"
      android:layout_height="200dp"
      android:src="@drawable/manga_logo"
      />

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="@string/app_name"
```

```
        android:textColor="@color/white"
        android:textSize="36sp"
        android:textStyle="bold"
        android:shadowColor="@color/black"
        android:fontFamily="serif"
        android:shadowDx="2"
        android:shadowDy="2"
        android:shadowRadius="3"
        android:background="@color/black"
        android:padding="8dp"/>
</LinearLayout>
```

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg002"
    android:padding="20dp"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/iconTv"
        android:src="@drawable/manga_logo"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/iconTv"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:text="Welcome to Manga Book App"
        android:fontFamily="serif"
        android:textColor="@color/black"
        android:textSize="16sp"
        android:textStyle="bold"/>

    <Button
        android:id="@+id/loginadminbtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@id/loginusrbtn"
        android:layout_marginBottom="10dp"
        android:backgroundTint="#FE2F52"
        android:text="Login as Admin"
        android:textColor="@color/white"
        app:cornerRadius="50dp"
        tools:ignore="ExtraText" />

    <Button
        android:id="@+id/loginusrbtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_above="@id/skipbtn"
        android:layout_marginBottom="10dp"
        android:backgroundTint="#FE2F52"
        android:text="Login as User"
        android:textColor="@color/white"
        app:cornerRadius="50dp"
        tools:ignore="ExtraText" />

    <Button
        android:id="@+id/skipbtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:text="Continue Without Login"
        android:backgroundTint="#FE2F52"
        android:textColor="@color/white"
        app:cornerRadius="50dp"
        tools:ignore="ExtraText"/>

</RelativeLayout>
```

Activity_login_admin.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg003"
    tools:context=".LoginActivity">

    <RelativeLayout
        android:id="@+id/toolbarRL"
        android:layout_width="match_parent"
        android:layout_height="250dp"
        android:background="@drawable/shape_toolbar01">

        <ImageView
            android:id="@+id/iconTv"
            android:layout_width="118dp"
            android:layout_height="118dp"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="48dp"
            android:src="@drawable/manga_logo"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/iconTv"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="18dp"
            android:text="Please Login"
            android:textColor="@color/black"
            android:textStyle="bold"
            android:fontFamily="serif"
            android:textSize="28dp"/>

    </RelativeLayout>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="30dp"
        android:layout_above="@+id/noAccountTv"
        android:layout_marginEnd="30dp"
        android:layout_marginTop="225dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <!--Email-->
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/emailTil"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                app:hintEnabled="false">

                <EditText
                    android:id="@+id/emailEt"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:drawableStart="@drawable/ic_email_gray"
                    android:background="@drawable/shape_edittext01"
                    android:drawablePadding="18dp"
                    android:hint="Email"
                    android:inputType="textEmailAddress"
                    android:padding="18dp"
                    android:textColorHint="@color/gray01"
                    android:textColor="@color/black"/>

            </com.google.android.material.textfield.TextInputLayout>

            <!--Password-->
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/passwordTil"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_below="@id/emailTil"
                android:layout_marginTop="5dp"
                app:passwordToggleEnabled="true"
                app:hintEnabled="false">

                <EditText
                    android:id="@+id/passwordEt"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:drawableStart="@drawable/ic_lock_gray"
                    android:background="@drawable/shape_edittext01"
                    android:drawablePadding="18dp"
                    android:hint="Password"
                    android:inputType="textPassword"
                    android:padding="18dp"
                    android:textColor="@color/black"
                    android:textColorHint="@color/gray01"/>

            </com.google.android.material.textfield.TextInputLayout>

            <!--Forgot Password-->
            <TextView
                android:id="@+id/forgotTv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@id/passwordTil"
                android:layout_alignParentEnd="true"
                android:layout_marginTop="5dp"
                android:padding="5dp"
                android:text="Forgot Password?"
                android:textColor="@color/gray02"
                android:textStyle="bold"/>

            <!--LoginButton-->
            <Button
                android:id="@+id/loginbtn"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_below="@+id/forgotTv"
                android:layout_marginTop="20dp"
                android:backgroundTint="@color/logo_clr"
```

```
                android:text="Login"
                android:textColor="@color/white"
                android:textStyle="bold"
                android:fontFamily="serif"
                app:cornerRadius="150sp"
                tools:ignore="ExtraText"/>

        </RelativeLayout>

    </ScrollView>

    <!--No Account;Register-->
    <TextView
        android:id="@+id/noAccountTv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:padding="18dp"
        android:text="@string/new_admin_signup"
        android:textColor="@color/black"
        android:fontFamily="serif"/>

</RelativeLayout>
```

Activity_register_admin.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg003"
    tools:context=".RegisterActivity">

    <RelativeLayout
        android:id="@+id/toolbarRL"
        android:layout_width="match_parent"
        android:layout_height="250dp"
        android:background="@drawable/shape_toolbar01">

        <ImageButton
            android:id="@+id/backBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/ic_back_white"
            android:background="@android:color/transparent"
            android:layout_margin="10dp"
            android:padding="5dp"/>
        <ImageView
            android:id="@+id/iconTv"
            android:layout_width="118dp"
            android:layout_height="118dp"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="48dp"
            android:src="@drawable/manga_logo"/>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/iconTv"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="18dp"
            android:text="Create New Account"
            android:textColor="@color/black"
            android:textStyle="bold"
            android:fontFamily="serif"
            android:textSize="28dp"/>

    </RelativeLayout>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="30dp"
        android:layout_marginEnd="30dp"
        android:layout_marginTop="225dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <!--Name-->
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/nameTil"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                app:hintEnabled="false">

                <EditText
                    android:id="@+id/nameEt"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:drawableStart="@drawable/ic_person_gray"
                    android:background="@drawable/shape_edittext01"
                    android:drawablePadding="18dp"
                    android:hint="Name"
                    android:inputType="textPersonName|textCapWords"
                    android:padding="18dp"
                    android:textColorHint="@color/gray01"
                    android:textColor="@color/black"/>

            </com.google.android.material.textfield.TextInputLayout>

            <!--Email-->
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/emailTil"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_below="@id/nameTil"
                android:layout_marginTop="5dp"
                app:hintEnabled="false">

                <EditText
                    android:id="@+id/emailEt"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:drawableStart="@drawable/ic_email_gray"
                    android:background="@drawable/shape_edittext01"
                    android:drawablePadding="18dp"
                    android:hint="Email"
                    android:inputType="textEmailAddress"
                    android:padding="18dp"
                    android:textColorHint="@color/gray01"
                    android:textColor="@color/black"/>

            </com.google.android.material.textfield.TextInputLayout>

            <!--Password-->
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/passwordTil"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_below="@id/emailTil"
                android:layout_marginTop="5dp"
                app:passwordToggleEnabled="true"
                app:hintEnabled="false">

                <EditText
                    android:id="@+id/passwordEt"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:drawableStart="@drawable/ic_lock_gray"
                    android:background="@drawable/shape_edittext01"
                    android:drawablePadding="18dp"
                    android:hint="Password"
                    android:inputType="textPassword"
                    android:padding="18dp"
```

```
                    android:textColor="@color/black"
                    android:textColorHint="@color/gray01"/>

            </com.google.android.material.textfield.TextInputLayout>

            <!--Confirm Password-->
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/cpasswordTil"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_below="@id/passwordTil"
                android:layout_marginTop="5dp"
                app:passwordToggleEnabled="true"
                app:hintEnabled="false">

                <EditText
                    android:id="@+id/cpasswordEt"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:drawableStart="@drawable/ic_lock_gray"
                    android:background="@drawable/shape_edittext01"
                    android:drawablePadding="18dp"
                    android:hint="Confirm Password"
                    android:inputType="textPassword"
                    android:padding="18dp"
                    android:textColor="@color/black"
                    android:textColorHint="@color/gray01"/>

            </com.google.android.material.textfield.TextInputLayout>

            <!--LoginButton-->
            <Button
                android:id="@+id/registerbtn"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_below="@+id/cpasswordTil"
                android:layout_marginTop="20dp"
                android:backgroundTint="@color/logo_clr"
                android:text="Register"
                android:textColor="@color/white"
                android:textStyle="bold"
                android:fontFamily="serif"
                app:cornerRadius="150sp"
                tools:ignore="ExtraText"/>

        </RelativeLayout>

    </ScrollView>

</RelativeLayout>
```

Activity_dashboard_admin.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg004"
    tools:context=".DashboardAdminActivity">

    <!--Toolbar-->
    <RelativeLayout
        android:id="@+id/toolbarRL"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:background="@drawable/shape_toolbar02"
        android:paddingStart="16dp"
        android:paddingEnd="16dp"
        android:paddingTop="35dp"
        android:elevation="4dp">

        <TextView
            android:id="@+id/titleTv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:layout_centerHorizontal="true"
            android:text="Dashboard Admin"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:fontFamily="serif"/>

        <TextView
            android:id="@+id/subTitleTv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:layout_centerHorizontal="true"
            android:layout_below="@+id/titleTv"
            android:text="abc@gmail.com"
            android:textColor="@color/black"/>

        <ImageButton
            android:id="@+id/logoutBtn"
            android:layout_width="50dp"
            android:layout_height="wrap_content"
```

```
                android:layout_alignParentEnd="true"
                android:layout_marginEnd="0dp"
                android:background="?attr/selectableItemBackground"
                android:clickable="true"
                android:contentDescription="Logout"
                android:focusable="true"
                android:padding="8dp"
                android:scaleType="centerInside"
                android:src="@drawable/ic_logout_white" />

        </RelativeLayout>

        <!--Search-->
        <EditText
            android:id="@+id/searchET"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/toolbarRL"
            android:hint="Search"
            android:layout_marginStart="10dp"
            android:layout_marginEnd="10dp"
            android:layout_marginTop="10dp"
            android:drawableStart="@drawable/ic_search_white"
            android:padding="10dp"
            android:background="@color/logo_clr"
            android:textColorHint="@color/gray01"/>

        <!--Categories-->
        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/categoriesRv"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:listitem="@layout/row_category"
            android:layout_margin="5dp"
            android:layout_below="@id/searchET"
            app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
            android:layout_above="@id/bottomLl"
            />

        <LinearLayout
            android:id="@+id/bottomLl"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentBottom="true"
            android:layout_marginStart="10dp"
            android:layout_marginEnd="10dp"
            android:orientation="horizontal">

            <Button
                android:id="@+id/addCategoryBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="+Add Category"
            android:layout_weight="1"
            android:layout_marginEnd="10dp"
            android:layout_marginTop="20dp"
            android:backgroundTint="@color/logo_clr"
            android:textColor="@color/white"
            android:textStyle="bold"
            android:fontFamily="serif"
            app:cornerRadius="150sp"
            tools:ignore="ExtraText"/>

        <com.google.android.material.floatingactionbutton.FloatingActionButton
            android:id="@+id/addPdfFab"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/ic_pdf_add_white"
            app:backgroundTint="@color/logo_clr"
            app:fabCustomSize="50dp"/>


    </LinearLayout>

</RelativeLayout>
```

Activity_pdf_add.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg007"
    tools:context=".PdfAddActivity">

    <!--Toolbar-->
    <RelativeLayout
        android:id="@+id/toolbarRL"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/shape_toolbar01">

        <ImageButton
            android:id="@+id/backBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:background="@android:color/transparent"
        android:layout_centerVertical="true"
        android:padding="5dp"
        android:src="@drawable/ic_back_white"/>

    <ImageView
        android:id="@+id/iconTv"
        android:layout_width="110dp"
        android:layout_height="110dp"
        android:layout_centerHorizontal="true"
        android:layout_below="@+id/backBtn"
        android:src="@drawable/manga_logo"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/iconTv"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:fontFamily="serif"
        android:text="Add a New Book/Manga"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:textSize="28sp"/>

    <ImageButton
        android:id="@+id/attachBtn"
        android:layout_width="67dp"
        android:layout_height="120dp"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="10dp"
        android:background="@android:color/transparent"
        android:padding="5dp"
        android:src="@drawable/ic_attach_white" />

</RelativeLayout>

<!--Input Title-->

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/titleTil"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="30dp"
    android:layout_marginTop="225dp"
    android:layout_marginEnd="30dp"
    app:hintEnabled="false">

    <EditText
        android:id="@+id/titleEt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/shape_edittext01"
        android:drawableStart="@drawable/ic_pdf_white"
        android:drawablePadding="10dp"
        android:hint="Book Title"
        android:inputType="textPersonName|textCapWords"
        android:padding="10dp"
        android:textColor="@color/black"
        android:textSize="16sp"
        android:textColorHint="@color/gray01"/>

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/descriptionTil"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="30dp"
    android:layout_marginTop="10dp"
    android:layout_below="@+id/titleTil"
    android:layout_marginEnd="30dp"
    app:hintEnabled="false">

    <EditText
        android:id="@+id/descriptionEt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/shape_edittext01"
        android:drawableStart="@drawable/ic_pdf_white"
        android:drawablePadding="10dp"
        android:hint="Book Description"
        android:inputType="textMultiLine|textCapSentences"
        android:padding="10dp"
        android:textSize="16sp"
        android:textColor="@color/black"
        android:textColorHint="@color/gray01"/>

</com.google.android.material.textfield.TextInputLayout>

<!--Pick Category from Dialog-->
<TextView
    android:id="@+id/categoryTv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/descriptionTil"
```

```
        android:layout_marginStart="30dp"
        android:layout_marginEnd="30dp"
        android:layout_marginTop="10dp"
        android:background="@drawable/shape_edittext01"
        android:drawableStart="@drawable/ic_category_white"
        android:drawableEnd="@drawable/ic_down_white"
        android:drawablePadding="10dp"
        android:gravity="center_vertical"
        android:padding="10dp"
        android:hint="Book Category"
        android:textColorHint="@color/gray01"
        android:textColor="@color/black"
        android:textSize="16sp"/>

    <Button
        android:id="@+id/submitBtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/categoryTv"
        android:layout_marginStart="30dp"
        android:layout_marginEnd="30dp"
        android:layout_marginTop="10dp"
        android:backgroundTint="@color/logo_clr"
        android:text="Upload"
        android:textColor="@color/black"
        android:textStyle="bold"
        android:fontFamily="serif"
        app:cornerRadius="150sp"
        tools:ignore="ExtraText"/>

</RelativeLayout>
```

Activity_pdf_list_admin.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg009"
    tools:context=".PdfListAdminActivity">

    <!--Toolbar-->
    <RelativeLayout
        android:id="@+id/toolbarRl"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:padding="20dp"
        android:paddingStart="16dp"
    android:paddingEnd="36dp"
    android:paddingTop="35dp"
    android:background="@drawable/shape_toolbar02">

    <ImageButton
        android:id="@+id/backBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_back_white"
        android:layout_centerVertical="true"
        android:layout_marginStart="5dp"
        android:background="@android:color/transparent"/>

    <TextView
        android:id="@+id/titleTv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Books/Manga"
        android:layout_marginTop="8dp"
        android:textStyle="bold"
        android:layout_centerHorizontal="true"
        android:textColor="@color/black"/>

    <TextView
        android:id="@+id/subtitleTv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Category"
        android:textColor="@color/black"
        android:layout_centerHorizontal="true"
        android:layout_below="@id/titleTv"/>

</RelativeLayout>

<EditText
    android:id="@+id/searchEt"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:layout_marginStart="10dp"
    android:layout_marginEnd="10dp"
    android:background="@color/white"
    android:layout_below="@id/toolbarRl"
    android:padding="10dp"
    android:hint="Search"
    android:textColorHint="@color/gray02"
    android:drawableStart="@drawable/ic_search_gray"
    android:drawablePadding="10dp"
    android:textColor="@color/black"/>
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/bookRv"
        android:layout_width="wrap_content"
        android:layout_below="@+id/searchEt"
        android:layout_height="match_parent"
        android:layout_margin="10dp"
        app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
        tools:listitem="@layout/row_pdf_admin"/>

</RelativeLayout>
```

Acitivity_category_add.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg005"
    tools:context=".CategoryAddActivity">

  <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="250dp"
      android:background="@drawable/shape_toolbar01">

    <ImageButton
        android:id="@+id/backBtn"
        android:layout_width="59dp"
        android:layout_height="128dp"
        android:layout_marginStart="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="10dp"
        android:background="@android:color/transparent"
        android:padding="5dp"
        android:src="@drawable/ic_back_white" />

    <ImageView
        android:id="@+id/iconTv"
        android:layout_width="110dp"
        android:layout_height="110dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="48dp"
        android:src="@drawable/manga_logo"/>
  <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@+id/iconTv"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="10dp"
      android:fontFamily="serif"
      android:text="Add a new Category"
      android:textStyle="bold"
      android:textColor="@color/black"
      android:textSize="20dp"/>

</RelativeLayout>

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/categoryTil"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:hintEnabled="false"
    android:layout_marginStart="30dp"
    android:layout_marginEnd="30dp"
    android:layout_marginTop="225dp">

  <EditText
      android:id="@+id/categoryET"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:background="@drawable/shape_edittext01"
      android:drawableStart="@drawable/ic_category_white"
      android:drawablePadding="10dp"
      android:hint="Category Title"
      android:inputType="textPersonName|textCapWords"
      android:padding="10dp"
      android:textColor="@color/black"
      android:textColorHint="@color/gray01"/>

</com.google.android.material.textfield.TextInputLayout>

<Button
    android:id="@+id/submitBtn"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="30dp"
    android:layout_marginEnd="20dp"
    android:layout_below="@+id/categoryTil"
    android:text="Submit"
    android:textColor="@color/white"
    android:backgroundTint="@color/logo_clr"
    android:textStyle="bold"
    android:fontFamily="serif"
```

```
      app:cornerRadius="150sp"
      tools:ignore="ExtraText"/>


</RelativeLayout>
```

Row_category.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    app:contentPadding="10dp"
    app:cardCornerRadius="8dp"
    app:cardElevation="8dp"
    app:cardUseCompatPadding="true"
    app:cardBackgroundColor="@color/white">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_vertical"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/categoryTv"
            android:layout_width="8dp"
            android:layout_height="wrap_content"
            android:text="Category Title"
            android:layout_weight="1"
            android:textColor="@color/black"/>

        <ImageButton
            android:id="@+id/deleteBtn"
            android:layout_width="48dp"
            android:layout_height="48dp"
            android:src="@drawable/ic_delete_white"
            android:background="@drawable/shape_button03"/>

    </LinearLayout>

</androidx.cardview.widget.CardView>
```

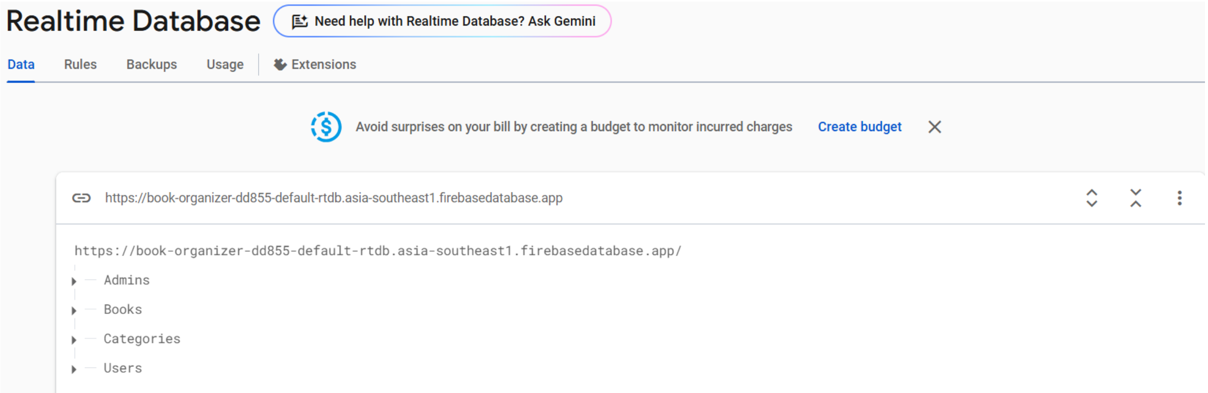
Row_pdf_admin:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
```

```
android:layout_height="140dp"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_marginBottom="5dp"
app:cardBackgroundColor="@color/white"
app:cardCornerRadius="10dp"
app:cardElevation="0dp"
app:contentPadding="5dp">

<RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <!--PdfView-->
    <RelativeLayout
        android:id="@+id/pdfRl"
        android:layout_width="100dp"
        android:layout_height="148dp"
        android:layout_marginEnd="5dp"
        android:background="@color/gray02">

        <com.github.barteksc.pdfviewer.PDFView
            android:id="@+id/pdfView"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>


        <ProgressBar
            android:id="@+id/progressBar"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>

    </RelativeLayout>

    <TextView
        android:id="@+id/titleTv"
        style="@style/TextAppearance.MaterialComponents.Headline6"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_toStartOf="@id/moreBtn"
        android:text="Book Title"
        android:textColor="@color/black"
        android:layout_toEndOf="@id/pdfRl"
        android:maxLines="1"/>

    <TextView
        android:id="@+id/descriptionTv"
        style="@style/TextAppearance.MaterialComponents.Body2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Book Description"
        android:textColor="@color/black"
        android:layout_toEndOf="@id/pdfRl"
        android:layout_marginTop="8dp"
        android:layout_below="@id/titleTv"
        android:maxLines="4"
        android:ellipsize="end"/>

    <TextView
        android:id="@+id/categoryTv"
        style="@style/TextAppearance.MaterialComponents.Caption"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Category"
        android:textColor="@color/black"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:maxLines="1"/>

    <TextView
        android:id="@+id/sizeTv"
        style="@style/TextAppearance.MaterialComponents.Caption"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="13 MB"
        android:textColor="@color/black"
        android:layout_toEndOf="@id/pdfRl"
        android:layout_alignParentBottom="true"
        android:maxLines="1"/>

    <TextView
        android:id="@+id/dateTv"
        style="@style/TextAppearance.MaterialComponents.Caption"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="dd/MM/yyyy"
        android:textColor="@color/black"
        android:layout_toEndOf="@id/sizeTv"
        android:layout_toStartOf="@id/categoryTv"
        android:textAlignment="center"
        android:layout_alignParentBottom="true"
        android:maxLines="1"/>

    <ImageButton
        android:id="@+id/moreBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@android:color/transparent"
        android:padding="5dp"
        android:outlineAmbientShadowColor="@color/black"
        android:src="@drawable/ic_more_black"
```

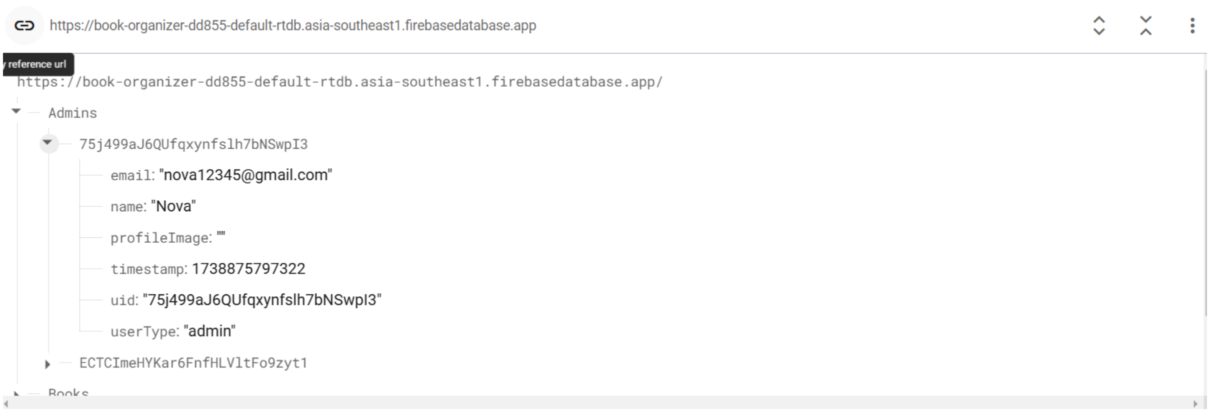
```
        android:layout_alignParentEnd="true"/>

    </RelativeLayout>

</androidx.cardview.widget.CardView>
```

Database:



Realtime Database
Need help with Realtime Database? Ask Gemini
Data
Rules
Backups
Usage
Extensions
Avoid surprises on your bill by creating a budget to monitor incurred charges
Create budget
https://book-organizer-dd855-default-rtdb.asia-southeast1.firebasedatabase.app

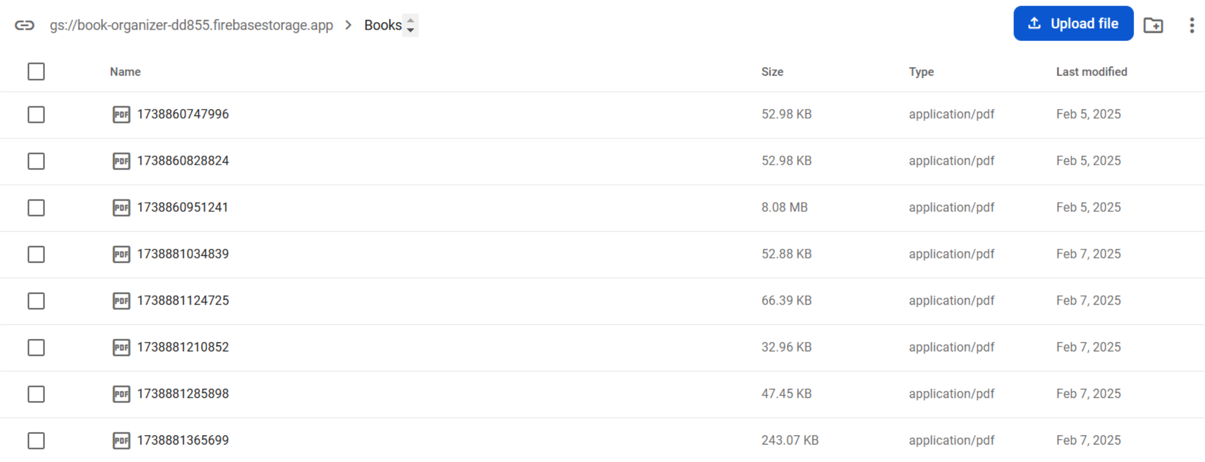
```
https://book-organizer-dd855-default-rtdb.asia-southeast1.firebasedatabase.app/
  ▸ — Admins
  ▸ — Books
  ▸ — Categories
  ▸ — Users
```



https://book-organizer-dd855-default-rtdb.asia-southeast1.firebasedatabase.app
y reference url

```
https://book-organizer-dd855-default-rtdb.asia-southeast1.firebasedatabase.app/
▾ — Admins
    ▾ — 75j499aJ6QUfqxynfslh7bNSwpI3
          email: "nova12345@gmail.com"
          name: "Nova"
          profileImage: ""
          timestamp: 1738875797322
          uid: "75j499aJ6QUfqxynfslh7bNSwpI3"
          userType: "admin"
    ▸ — ECTCImeHYKar6FnfHLVltFo9zyt1
▸ — Books
```

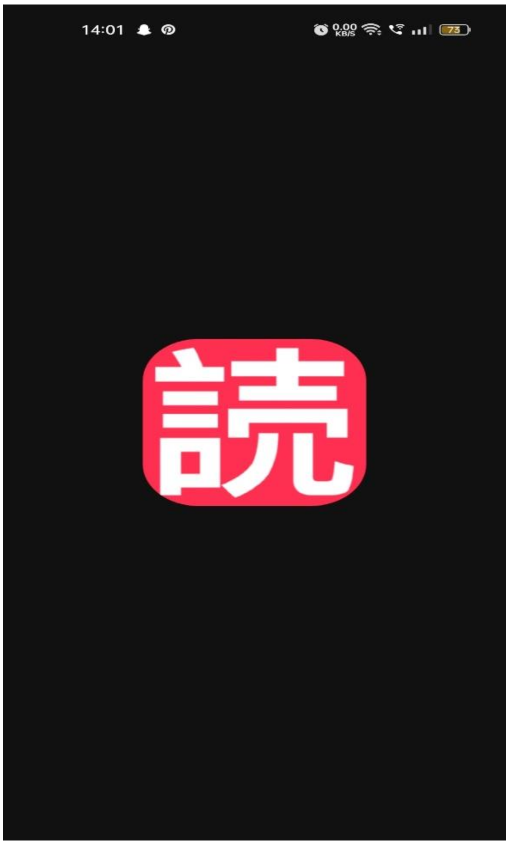
Storage:

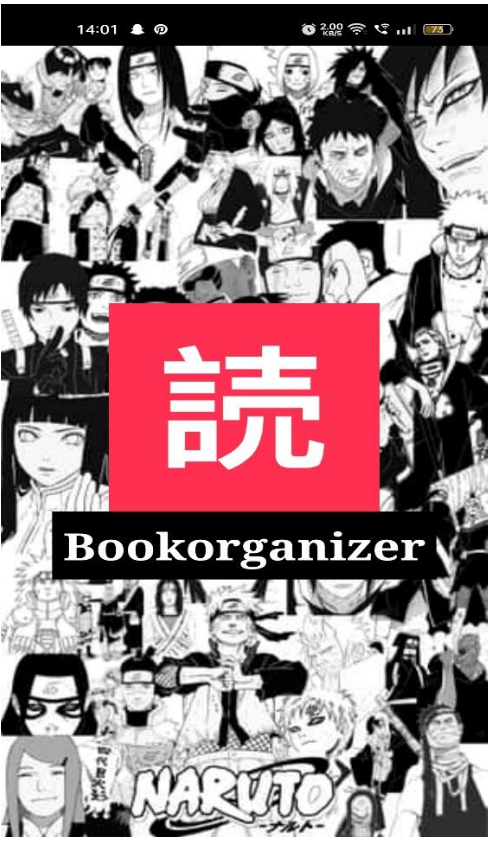


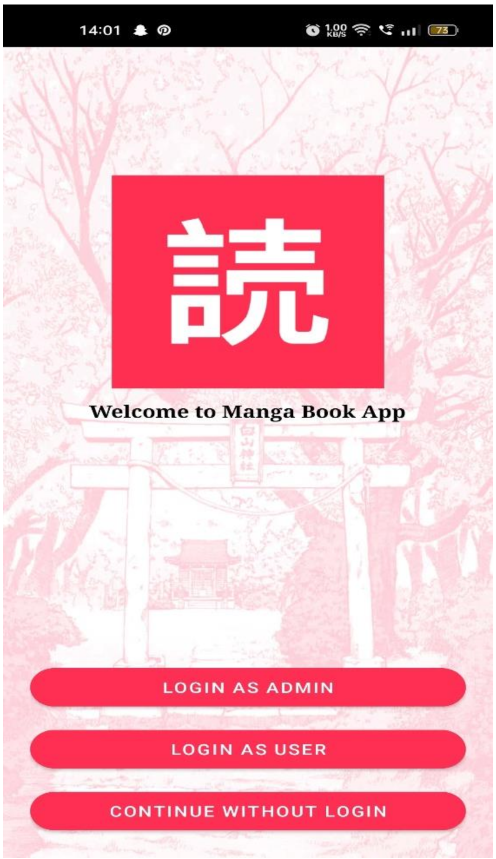
□	Name	Size	Type	Last modified
□	1738860747996	52.98 KB	application/pdf	Feb 5, 2025
□	1738860828824	52.98 KB	application/pdf	Feb 5, 2025
□	1738860951241	8.08 MB	application/pdf	Feb 5, 2025
□	1738881034839	52.88 KB	application/pdf	Feb 7, 2025
□	1738881124725	66.39 KB	application/pdf	Feb 7, 2025
□	1738881210852	32.96 KB	application/pdf	Feb 7, 2025
□	1738881285898	47.45 KB	application/pdf	Feb 7, 2025
□	1738881365699	243.07 KB	application/pdf	Feb 7, 2025

Output:

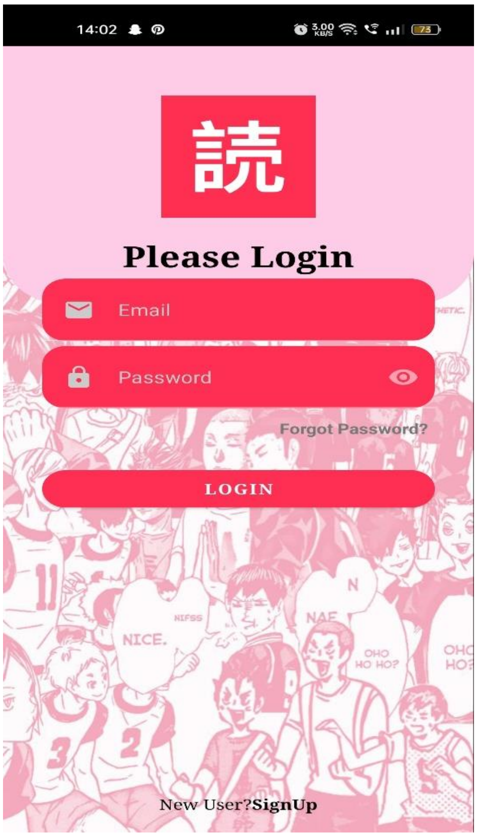
First and Splash Screen:

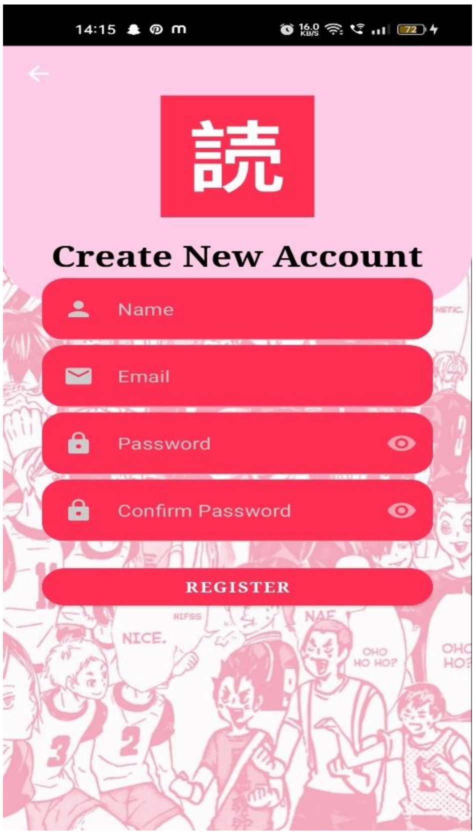




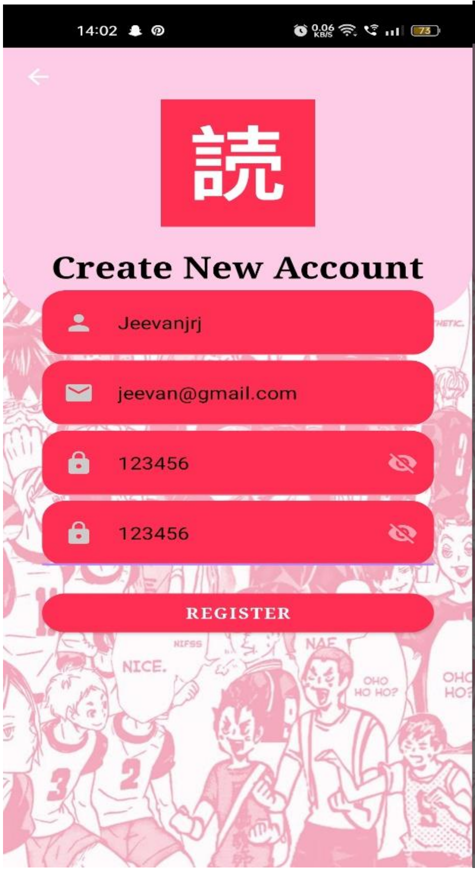


Login and Register page(Before entering credentials):





After Entering the Credentials:

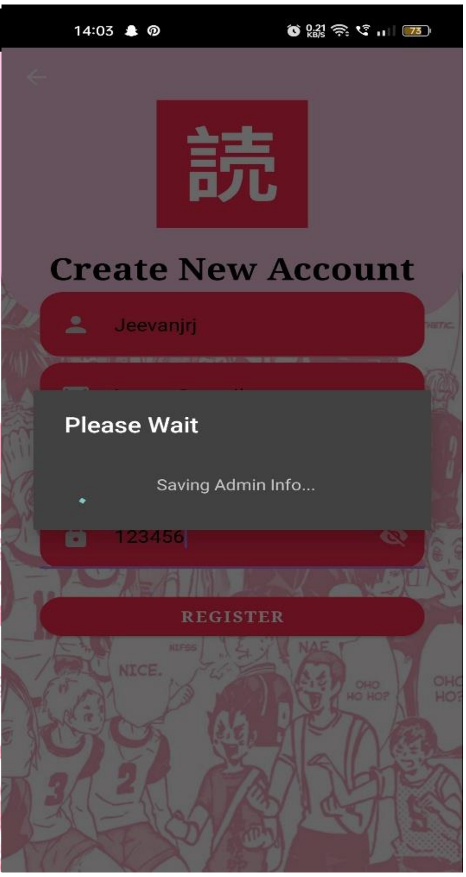


Jeevanjrj

jeevan@gmail.com

123456

123456

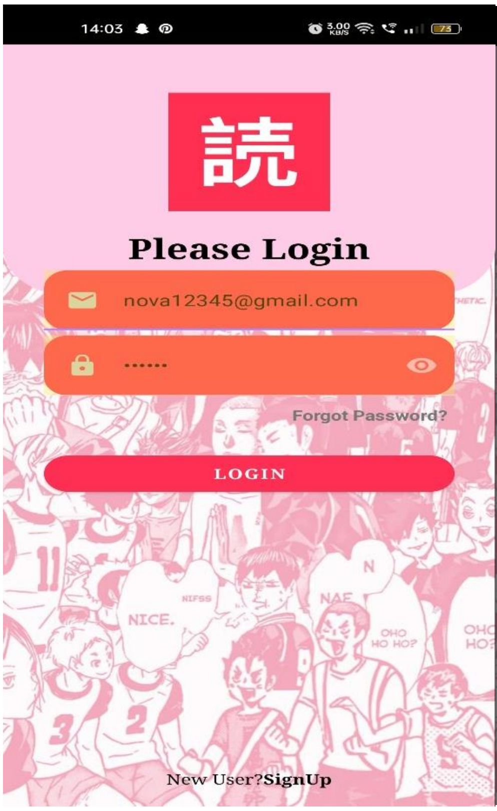


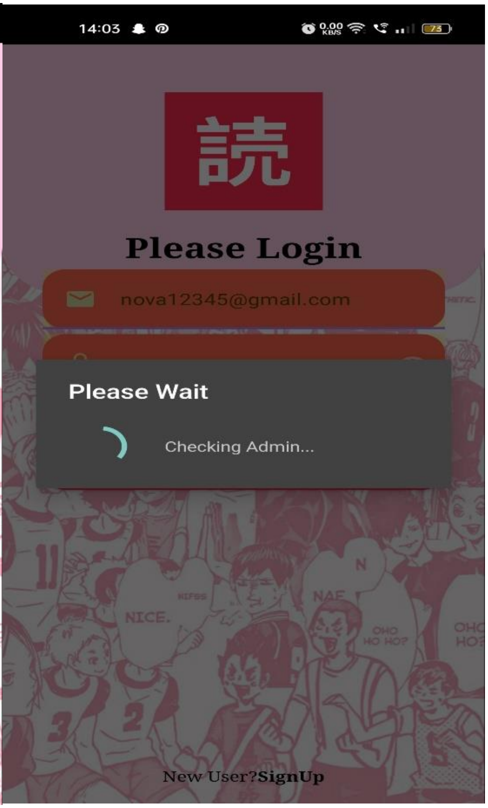
Jeevanjrj

Please Wait

Saving Admin Info...

123456

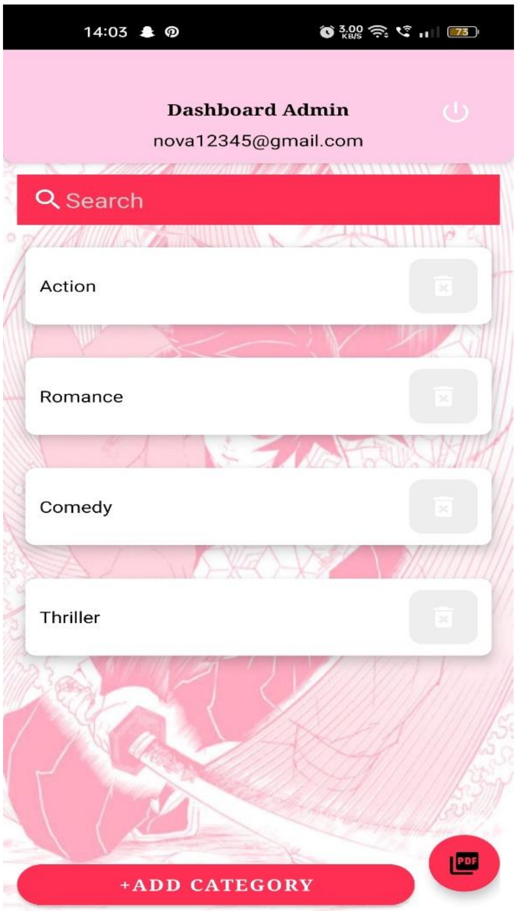


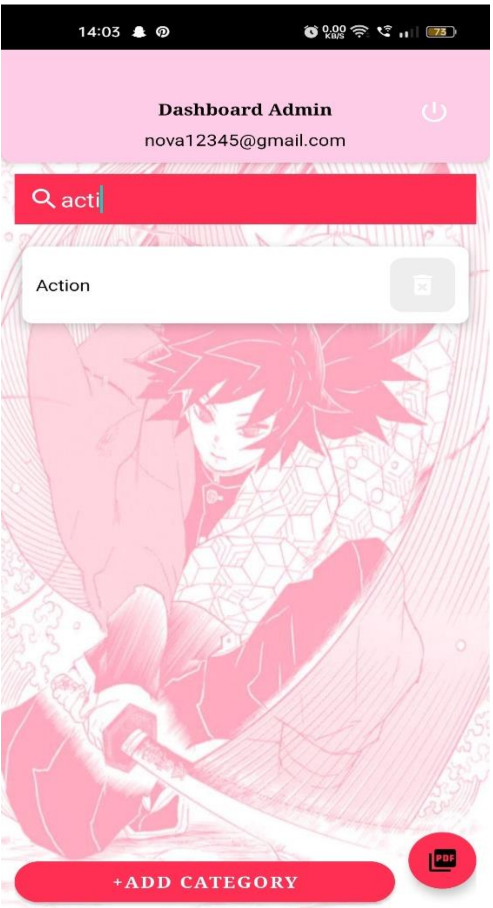


Please Wait

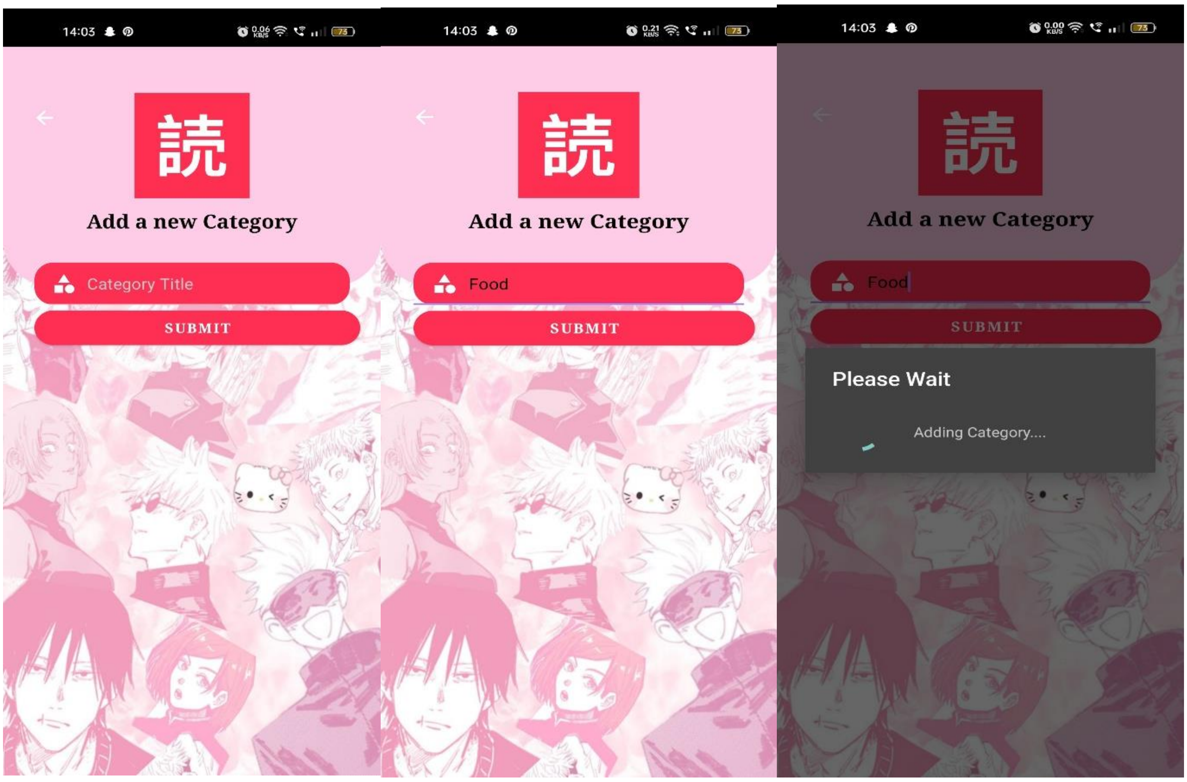
Checking Admin...

After Logging we will be to the Admin Dashboard and there we can search for any category and we can find that particular category getting filtered out:





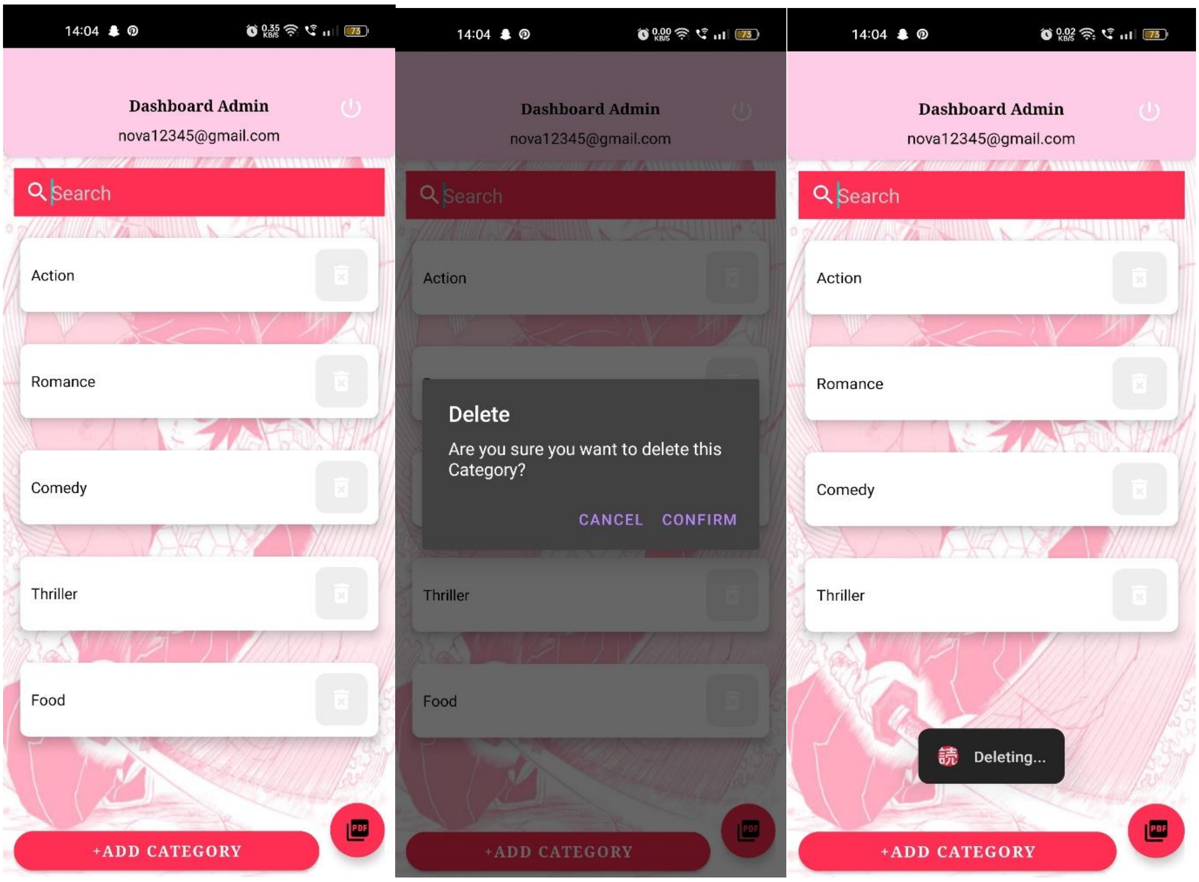
If we want to add the Category:



Please Wait

Adding Category....

After the Category gets added if we want to delete a particular category:



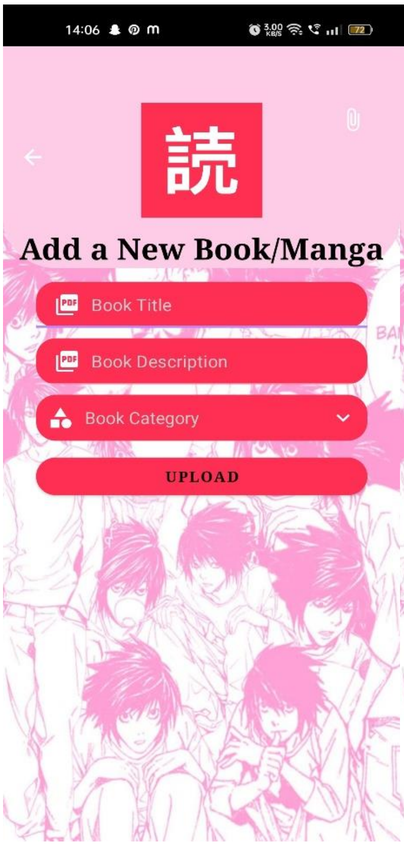
Delete

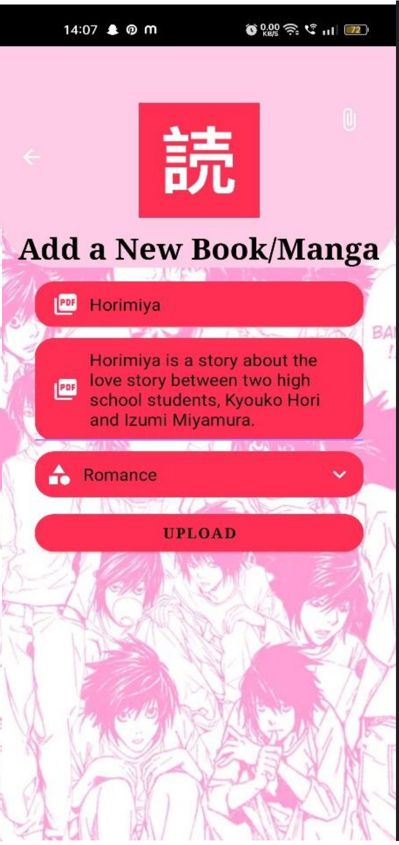
Are you sure you want to delete this Category?

CANCEL CONFIRM

Deleting...

If we want add new Manga Book Pdf from our Phone’s Storage:

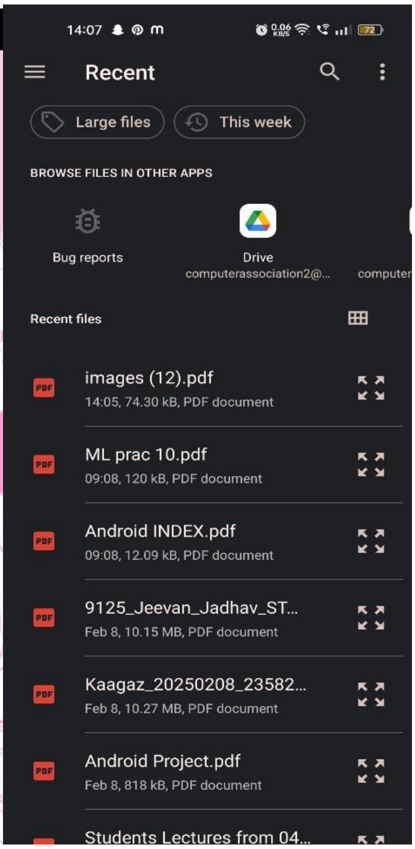




Horimiya

Horimiya is a story about the love story between two high school students, Kyouko Hori and Izumi Miyamura.

Romance



images (12).pdf
14:05, 74.30 kB, PDF document

ML prac 10.pdf
09:08, 120 kB, PDF document

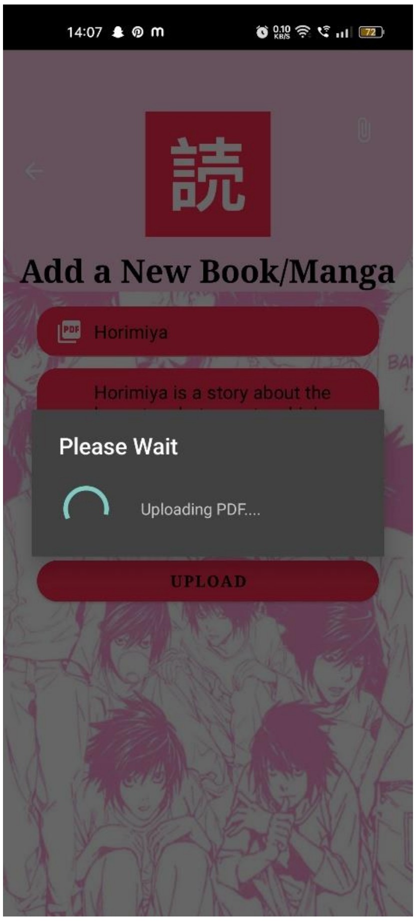
Android INDEX.pdf
09:08, 12.09 kB, PDF document

9125_Jeevan_Jadhav_ST...
Feb 8, 10.15 MB, PDF document

Kaagaz_20250208_23582...
Feb 8, 10.27 MB, PDF document

Android Project.pdf
Feb 8, 818 kB, PDF document

Students Lectures from 04...



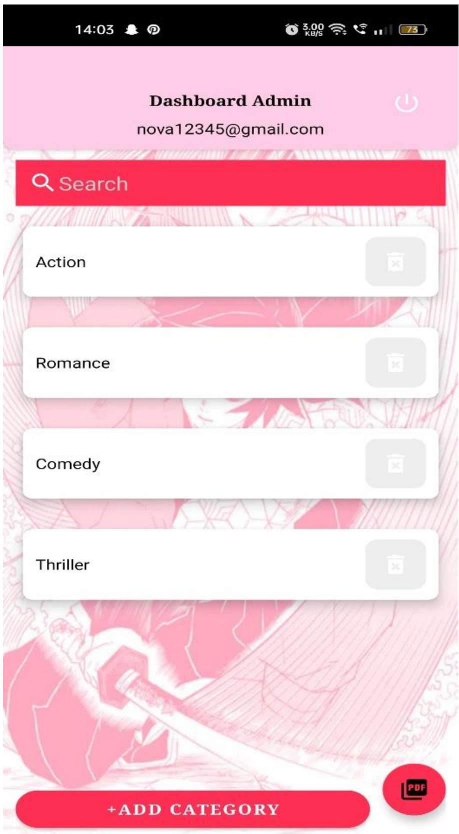
Horimiya

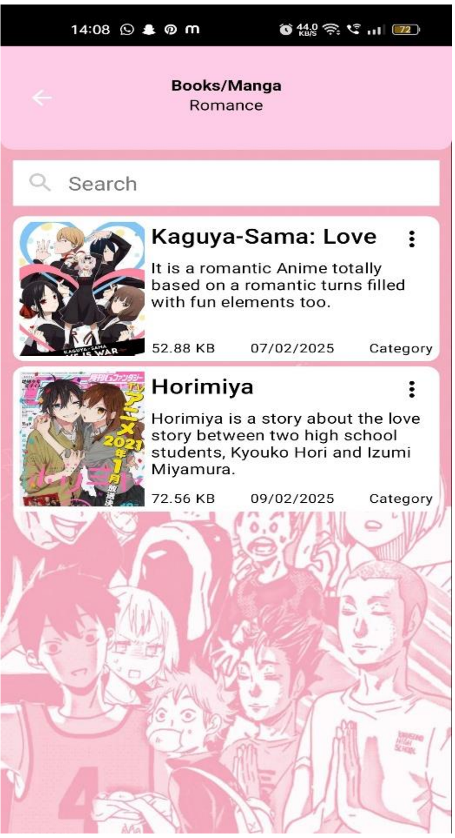
Horimiya is a story about the

Please Wait

Uploading PDF....

Now when we will select “Romance” Category then we can see our pdf is been added to the List.





Kaguya-Sama: Love

It is a romantic Anime totally based on a romantic turns filled with fun elements too.

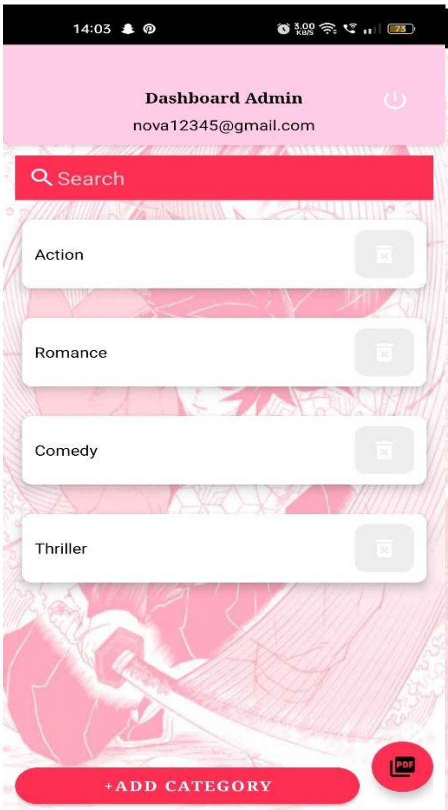
52.88 KB 07/02/2025 Category

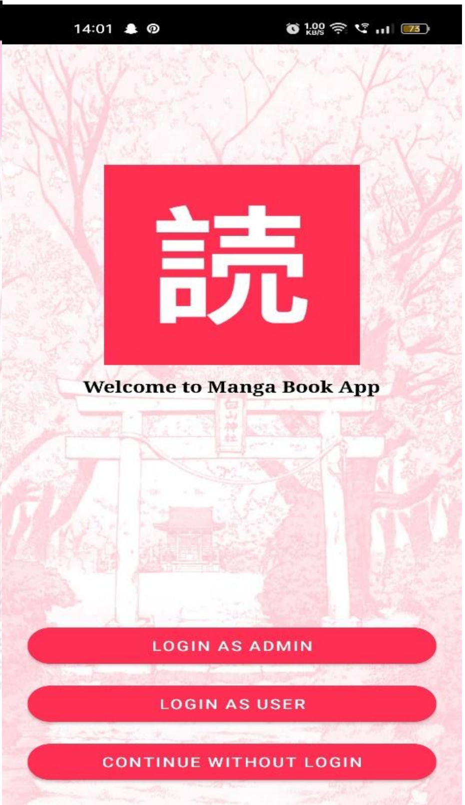
Horimiya

Horimiya is a story about the love story between two high school students, Kyouko Hori and Izumi Miyamura.

72.56 KB 09/02/2025 Category

Now we will logout from the dashboard by click the Logout button and we will be redirected to the MainActivity Page:





Conclusion:

The **Manga Book Organizer App** successfully provides a structured and efficient way to manage and organize manga collections. With its user-friendly interface, secure authentication system, and well-integrated features like category-based organization, search functionality, and PDF viewing, the app ensures a seamless experience for both admins and users. The use of **Firebase** for real-time data storage enhances performance, making book management more convenient and efficient.

By leveraging **Java (Android)** and **Android Studio**, the app achieves a robust and scalable design, meeting the objectives of easy book categorization, secure login, and smooth navigation. The ability to add, filter, and delete manga books ensures flexibility in managing collections. Overall, the **Manga Book Organizer App** is a practical and effective solution for manga enthusiasts, providing a well-structured approach to storing and accessing their favorite titles.

SWOT Analysis:

Strengths:

- **User-Friendly Interface:** The app provides a simple and intuitive UI, making navigation easy for users.
- **Efficient Book Management:** Users can add, categorize, search, and view manga books efficiently.
- **Real-Time Data Storage:** Firebase integration ensures instant updates and synchronization of data.
- **Secure Authentication:** Admin login ensures controlled access and prevents unauthorized modifications.
- **Lightweight and Fast Performance:** Optimized to run smoothly on various Android devices.
- **Offline Accessibility:** Cached data allows users to access previously loaded content even without an internet connection.

Weaknesses:

- **Dependent on Internet Connectivity:** Requires internet access for real-time database updates.
- **Limited File Format Support:** Only supports PDF format for manga books, restricting other formats like EPUB or CBZ.
- **No Multi-User Role Management:** Currently, only an admin can manage categories and book uploads.
- **No In-App Manga Reader Enhancements:** Lacks advanced reading features such as bookmarking, night mode, or adjustable text size.

Opportunities:

- **Expansion to Other Book Types:** The app can be expanded to support novels, comics, and textbooks.
- **Multi-User Role Implementation:** Introducing different user roles (Admin, Editor, Reader) for better management.
- **Offline Mode Enhancement:** Enable full offline access with local database storage.
- **Monetization Opportunities:** Can be monetized via ads, premium subscriptions, or partnerships.

Threats:

- **Competition from Similar Apps:** Other book organizer apps with more features may attract users.
- **Security Concerns:** Potential risks of data breaches or unauthorized access to user information.
- **Changing User Preferences:** Users may shift to other platforms that offer better reading experiences and additional features.
- **Dependence on Firebase:** Any service disruption in Firebase may affect data storage and app functionality.