BornToDeathSchemes/

│ settings.gradle

│ build.gradle

│ .gitignore

│

├── .github/

│ └── workflows/

│ └── build.yml

│

├── app/

│ ├── build.gradle

│ └── src/

│ └── main/

│ ├── AndroidManifest.xml

│ ├── java/com/example/childlifecycle/

│ │ ├── App.kt

│ │ ├── MainActivity.kt

│ │ ├── SchemeListActivity.kt

│ │ ├── SchemeDetailActivity.kt

│ │ ├── PDFViewerActivity.kt

│ │ ├── TrialManager.kt

│ │ ├── LoginActivity.kt

│ │ ├── MyFirebaseMessagingService.kt

│ │ ├── network/

│ │ │ ├── NetworkModule.kt

│ │ │ └── BackendApi.kt

│ │ └── data/

│ │ ├── AppDatabase.kt

│ │ ├── Scheme.kt

│ │ ├── Favorite.kt

│ │ └── SchemeDao.kt

│ └── res/

│ ├── layout/

│ │ ├── activity\_main.xml

│ │ ├── activity\_scheme\_list.xml

│ │ └── item\_scheme.xml

│ ├── values/strings.xml

│ └── values-hi/strings.xml

rootProject.name = "BornToDeathSchemes"

include ':app'

**build.gradle**

buildscript {

ext.kotlin\_version = '1.9.0'

repositories {

google()

mavenCentral()

}

dependencies {

classpath 'com.android.tools.build:gradle:8.1.0'

classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin\_version"

classpath 'com.google.gms:google-services:4.3.15'

}

}

allprojects { repositories { google(); mavenCentral() } }

**.gitignore**

.gradle/

build/

local.properties

.idea/

\*.iml

app/build/

\*.jks

name: Build Debug APK

on: [push]

jobs:

build:

runs-on: ubuntu-latest

steps:

- uses: actions/checkout@v4

- name: Set up JDK

uses: actions/setup-java@v4

with:

distribution: temurin

java-version: 17

- name: Make gradlew executable

run: chmod +x ./gradlew

- name: Build Debug APK

run: ./gradlew assembleDebug --no-daemon

- name: Upload APK

uses: actions/upload-artifact@v4

with:

name: app-debug

path: app/build/outputs/apk/debug/\*.apk

plugins {

id 'com.android.application'

id 'org.jetbrains.kotlin.android'

}

android {

namespace 'com.example.childlifecycle'

compileSdk 34

defaultConfig {

applicationId "com.example.childlifecycle"

minSdk 24

targetSdk 34

versionCode 1

versionName "0.1"

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'

}

}

buildFeatures { viewBinding true }

}

dependencies {

implementation "org.jetbrains.kotlin:kotlin-stdlib:1.9.0"

implementation 'androidx.core:core-ktx:1.10.0'

implementation 'androidx.appcompat:appcompat:1.7.0'

implementation 'com.google.android.material:material:1.9.0'

implementation 'androidx.constraintlayout:constraintlayout:2.1.4'

implementation 'androidx.recyclerview:recyclerview:1.3.0'

implementation 'androidx.room:room-runtime:2.5.2'

kapt 'androidx.room:room-compiler:2.5.2'

implementation 'androidx.room:room-ktx:2.5.2'

implementation 'com.squareup.retrofit2:retrofit:2.9.0'

implementation 'com.squareup.retrofit2:converter-moshi:2.9.0'

implementation 'androidx.work:work-runtime-ktx:2.8.1'

implementation 'com.google.firebase:firebase-messaging:23.2.1'

implementation 'com.razorpay:checkout:1.7.58'

implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.7.1'

}

<manifest package="com.example.childlifecycle"

xmlns:android="http://schemas.android.com/apk/res/android">

<uses-permission android:name="android.permission.INTERNET" />

<application

android:name=".App"

android:allowBackup="false"

android:label="@string/app\_name"

android:theme="@style/Theme.AppCompat.Light.NoActionBar">

<activity android:name=".PDFViewerActivity" android:exported="false" />

<activity android:name=".SchemeDetailActivity" android:exported="false" />

<activity android:name=".SchemeListActivity" android:exported="true" />

<activity android:name=".MainActivity" android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<service android:name=".MyFirebaseMessagingService" android:exported="false">

<intent-filter>

<action android:name="com.google.firebase.MESSAGING\_EVENT" />

</intent-filter>

</service>

</application>

</manifest>

package com.example.childlifecycle

import android.app.Application

import androidx.room.Room

import com.example.childlifecycle.data.AppDatabase

class App : Application() {

companion object {

lateinit var instance: App

private set

}

lateinit var db: AppDatabase

private set

override fun onCreate() {

super.onCreate()

instance = this

db = Room.databaseBuilder(applicationContext, AppDatabase::class.java, "schemes-db")

.fallbackToDestructiveMigration()

.build()

}

}

package com.example.childlifecycle

import android.content.Intent

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import com.example.childlifecycle.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

private lateinit var binding: ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

binding = ActivityMainBinding.inflate(layoutInflater)

setContentView(binding.root)

binding.startTrialBtn.setOnClickListener {

TrialManager.startTrialIfNeeded(this)

startActivity(Intent(this, SchemeListActivity::class.java))

}

}

}

package com.example.childlifecycle

import android.content.Intent

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import androidx.lifecycle.lifecycleScope

import androidx.recyclerview.widget.LinearLayoutManager

import com.example.childlifecycle.data.Scheme

import com.example.childlifecycle.databinding.ActivitySchemeListBinding

import kotlinx.coroutines.launch

class SchemeListActivity : AppCompatActivity() {

private lateinit var binding: ActivitySchemeListBinding

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

binding = ActivitySchemeListBinding.inflate(layoutInflater)

setContentView(binding.root)

binding.recyclerView.layoutManager = LinearLayoutManager(this)

lifecycleScope.launch {

val schemes = App.instance.db.schemeDao().findByAgeGender(0, "all")

binding.recyclerView.adapter = SchemeAdapter(schemes) { scheme ->

val intent = Intent(this@SchemeListActivity, SchemeDetailActivity::class.java)

intent.putExtra("scheme\_id", scheme.id)

startActivity(intent)

}

}

}

}

package com.example.childlifecycle

import android.content.Intent

import android.net.Uri

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import com.example.childlifecycle.data.Scheme

class SchemeDetailActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

val id = intent.getStringExtra("scheme\_id") ?: return

val scheme: Scheme = App.instance.db.schemeDao().getSchemeSync(id)

// Show scheme details (simplified)

title = scheme.title

// Open PDF

val pdfIntent = Intent(this, PDFViewerActivity::class.java)

pdfIntent.putExtra("pdf\_url", scheme.sourceUrl)

startActivity(pdfIntent)

}

}

package com.example.childlifecycle

import android.os.Bundle

import android.view.WindowManager

import android.webkit.WebView

import android.webkit.WebViewClient

import androidx.appcompat.app.AppCompatActivity

class PDFViewerActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

window.setFlags(WindowManager.LayoutParams.FLAG\_SECURE, WindowManager.LayoutParams.FLAG\_SECURE)

val webView = WebView(this)

webView.webViewClient = WebViewClient()

val url = intent.getStringExtra("pdf\_url")

webView.loadUrl("https://docs.google.com/gview?embedded=true&url=$url")

setContentView(webView)

}

}

package com.example.childlifecycle

import android.content.Context

import java.util.\*

object TrialManager {

private const val PREF = "trial\_prefs"

private const val KEY\_TRIAL\_START = "trial\_start"

fun startTrialIfNeeded(ctx: Context) {

val prefs = ctx.getSharedPreferences(PREF, Context.MODE\_PRIVATE)

if (!prefs.contains(KEY\_TRIAL\_START)) {

prefs.edit().putLong(KEY\_TRIAL\_START, System.currentTimeMillis()).apply()

}

}

fun isTrialActive(ctx: Context): Boolean {

val prefs = ctx.getSharedPreferences(PREF, Context.MODE\_PRIVATE)

val start = prefs.getLong(KEY\_TRIAL\_START, -1)

if (start == -1L) return false

val sevenDays = 7L \* 24 \* 3600 \* 1000

return System.currentTimeMillis() - start <= sevenDays

}

}

package com.example.childlifecycle.data

import androidx.room.Database

import androidx.room.RoomDatabase

@Database(entities = [Scheme::class, Favorite::class], version = 1)

abstract class AppDatabase : RoomDatabase() {

abstract fun schemeDao(): SchemeDao

}

package com.example.childlifecycle.data

import androidx.room.Entity

import androidx.room.PrimaryKey

@Entity(tableName = "schemes")

data class Scheme(

@PrimaryKey val id: String,

val title: String,

val description: String,

val sourceUrl: String,

val ageMin: Int?,

val ageMax: Int?,

val gender: String?

)

package com.example.childlifecycle.data

import androidx.room.Entity

import androidx.room.PrimaryKey

@Entity(tableName = "favorites")

data class Favorite(

@PrimaryKey val schemeId: String,

val savedAt: Long

)

package com.example.childlifecycle.data

import androidx.room.\*

@Dao

interface SchemeDao {

@Query("SELECT \* FROM schemes WHERE (ageMin IS NULL OR ageMin <= :age) AND (ageMax IS NULL OR ageMax >= :age) AND (gender IS NULL OR gender = 'all' OR gender = :gender)")

suspend fun findByAgeGender(age: Int, gender: String): List<Scheme>

@Insert(onConflict = OnConflictStrategy.REPLACE)

suspend fun insertAll(schemes: List<Scheme>)

@Query("SELECT \* FROM schemes WHERE id = :id")

fun getSchemeSync(id: String): Scheme

}

package com.example.childlifecycle.network

import retrofit2.Retrofit

import retrofit2.converter.moshi.MoshiConverterFactory

object NetworkModule {

private const val BASE = "https://example.com/api/" // Replace with real backend

val retrofit: Retrofit = Retrofit.Builder()

.baseUrl(BASE)

.addConverterFactory(MoshiConverterFactory.create())

.build()

val api: BackendApi = retrofit.create(BackendApi::class.java)

}

package com.example.childlifecycle.network

import com.example.childlifecycle.data.Scheme

import retrofit2.http.GET

interface BackendApi {

@GET("schemes")

suspend fun getSchemes(): List<Scheme>

}

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:gravity="center"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/startTrialBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/start\_trial" />

</LinearLayout>

**res/layout/activity\_scheme\_list.xml**

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

</LinearLayout>

**res/layout/item\_scheme.xml**

<TextView xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/schemeTitle"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="16dp"

android:textSize="18sp" />

**res/values/strings.xml**

<resources>

<string name="app\_name">Born to Death Schemes</string>

<string name="start\_trial">Start 7-day Free Trial</string>

<string name="view\_pdf">View Document (No Download)</string>

</resources>

**res/values-hi/strings.xml**

<resources>

<string name="app\_name">जन्म से मृत्यु तक योजनाएँ</string>

<string name="start\_trial">7 दिन की मुफ्त ट्रायल शुरू करें</string>

<string name="view\_pdf">दस्तावेज़ देखें (डाउनलोड नहीं)</string>

</resources>