Assignment 4
MAD
Siddhant Jain
Roll Number 22

STRING.XML

```
<resources>
    <string name="app_name">Exam Details</string>

    <!-- TODO: Remove or change this placeholder text -->
    <string name="hello_blank_fragment">Hello blank fragment</string>

    <!--    Exam Scheme-->
    <string name="exam_scheme_title">Youtube</string>
    <string name="credits_txt">Credits</string>
    <string name="credits_val">4</string>
    <string name="IA_txt">Mid Sem Uploading</string>
    <string name="IA_val">20</string>
    <string name="ISE_txt">End-Sem Examination</string>
    <string name="ISE_val">30</string>
    <string name="ESE_txt">Final Pratical</string>
    <string name="ESE_val">50</string>
    <string name="practical_txt">Written Exam</string>
    <string name="practical_val">30</string>
    <string name="presentation_txt">Presentation</string>
    <string name="presentation_val">20</string>
    <string name="total_txt">Total  </string>
    <string name="total_val">  150</string>



    <!--    Contents -->
    <string-array name="contents">
```

<item>Unit 1: Introduction to Youtube</item>
<item>Basics of lymorphism. Interfaces, collections and Exception handling</item>
<item>Unit 2: How to upload a youtube video</item>
<item>Introduction to the Android platform, Application framework, Basic Building blocks, Broadcast Receivers and
Content providers, UI Components, Intents and Intent Filters, Android API levels, AVD and the Android
Studio IDE.</item>
<item>Unit 3: Editing </item>
<item>UI Design: An among activities, Menus, Custom and compound Views, Notifications, UI
Events, Event Listeners.</item>
<item>Unit 4: Intent, Broadcast Receivers and Sensors </item>
<item>Explicit Intents, Implicit intents, Role of filters, Intent-matching rules, Filters in manifest and in dynamic
Broadcast Radcast. Sensors: Finding sensors, Accelerometers, </item>
<item>Unit 5: Google Ad Sense</item>
<item>Android File Syessing built in content providers, and Content provider
MIME types, searching, Adding, changing, and removing contents.</item>
<item>Unit 6: Termas and Policies</item>
<item>Introduction to Hybrid Applications, Benefits, Challenges, Power of Hybrid over Mobile Web, Architectural
approaches, Mobile Hybrid Application development framework.</item>
</string-array>

<!-- Practicals-->
<string name="practical_list_title">Practicals List</string>
<string-array name="practicals">
<item>Practical 1: Create a Video</item>
<item>Extend the programing written in practical no 1 and provide UI using android and run it in android virtual device.</item>
<item>Practical 2: Edit a Video</item>
<item>Create a Simple Calculator Android Application and run it in real device</item>
<item>Practical 3: Add effects in a video</item>
<item>Design a tab layout for android device and show course information in each tab. (Exam Scheme, Content, Practical List)</item>
<item>Practical 4: Uploading Videos</item>
<item>Extend the practical no. 4 and add list layout for different courses after clicking on course display the course information in tab layout as done in practical no. 4</item>
<item>Practical 5: Adding ads from Google Adsense</item>

```
        <item>For practical no. 5 provide the backend support for data storage and store all the course information in database, also provide a functionality to edit the course content in application.</item>
        <item>Practical 6: Hybrid App</item>
        <item>Create a hybrid app for the practical no.6 using any hybrid app framework</item>
    </string-array>

</resources>
```

ACTIVITY MAIN

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:background="@color/colorPrimary"
        android:layout_height="wrap_content">

        <TextView
            android:textColor="@color/text_color_dark"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/app_name"
            android:minHeight="?actionBarSize"

android:textAppearance="@style/TextAppearance.Design.CollapsingToolbar.Expanded"
            android:padding="15dp"
            />

        <com.google.android.material.tabs.TabLayout
```

```
        android:id="@+id/tablayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:tabGravity="fill"
        app:tabIndicatorColor="@color/tabIndicator"
        app:tabIndicatorHeight="2dp"
        app:tabMode="fixed"
        android:background="@color/colorPrimary"
        app:tabTextColor="@color/text_color_dark"
        />

    </com.google.android.material.appbar.AppBarLayout>

    <androidx.viewpager.widget.ViewPager
        android:id="@+id/myViewPager"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        >
    </androidx.viewpager.widget.ViewPager>
</LinearLayout>
```

MAIN ACTIVITY JAVA

```
package com.example.exams;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.viewpager.widget.ViewPager;

import android.os.Build;
import android.os.Bundle;
import android.widget.Toolbar;

import com.google.android.material.tabs.TabLayout;
```

```
public class MainActivity extends AppCompatActivity {

    private TabLayout tabLayout;
    private Toolbar toolbar;
    private ViewPager viewPager;
    private Toolbar supportActionBar;

    @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tabLayout = (TabLayout) findViewById(R.id.tablayout);
        viewPager = (ViewPager) findViewById(R.id.myViewPager);

        setUpViewPager(viewPager);
        tabLayout.setupWithViewPager(viewPager);
    }

    private void setUpViewPager(ViewPager viewPager) {
        com.example.exams.ViewPagerAdapter viewPagerAdapter = new
com.example.exams.ViewPagerAdapter(getSupportFragmentManager());
        viewPagerAdapter.addFragment(new ExamFragment(), "EXAM");
        viewPagerAdapter.addFragment(new ContentFragment(), "CONTENT");
        viewPagerAdapter.addFragment(new com.example.exams.PracticalFragment(),
"PRACTICAL");
        viewPager.setAdapter(viewPagerAdapter);
    }
}
```

SCREENSHOT

Content

Unit 1: Introduction to Youtube
Basics of java programing, string processing, multithreading, and input-output, object oriented concept: inheritance, encapsulation and polymorphism. Interfaces, collections and Exception handling
Unit 2: The Android Platform
Introduction to the Android platform, Application framework, Basic Building blocks, Broadcast Receivers and Content providers, UI Components, Intents and Intent Filters, Android API levels, AVD and the Android Studio IDE.
Unit 3: User Interface design
UI Design: Activities, Views, layouts and Common UI components, UI through code and XML, Activity lifecycle, Intents, communication among activities, Menus, Custom and compound Views, Notifications, UI Events, Event Listeners.
Unit 4: Intent, Broadcast Receivers and Sensors
Explicit Intents, Implicit intents, Role of filters, Intent-matching rules, Filters in manifest and in dynamic Broadcast Receivers, Creating Broadcast receiver, Receiving System Broadcast, Understanding Broadcast action, category and data, Sending Broadcast. Sensors: Finding sensors, Accelerometers,
Unit 5: Android Data Storage

Exam Details

Credits	4
Video Uploading	20
Effects Insertion	50
Effects Insertion	50
Practical	30
Presentation	20
	Total150

Practical

Practical 1: Tic Tac Toe Game

Extend the programing written in practical no 1 and provide UI using android and run it in android virtual device.

Practical 2: Calculator App

Create a Simple Calculator Android Application and run it in real device

Practical 3: Design Tab Layout

Design a tab layout for android device and show course information in each tab. (Exam Scheme, Content, Practical List)

Practical 4: Design List Layout

Extend the practical no. 4 and add list layout for different courses after clicking on course display the course information in tab layout as done in practical no. 4

Practical 5: Create Database Storage

For practical no. 5 provide the backend support for data storage and store all the course information in database, also provide a functionality to edit the course content in application.

Practical 6: Hybrid App

Create a hybrid app for the practical no.6 using any