**Program 11:**

**Hybrid App Development: Hybrid App vs Native App, React-Native, Flutter, Ionic, Xamarin**

**Task: Design Student Management App using any one of the Hybrid Frameworks or SDKs.**

Module: Hybrid Apps Vs. Native Apps

A hybrid app is an app created to run on all platforms. It combines elements of both native apps and web apps to produce this blended third option, written in a single codebase. Instead of rewriting apps in different operating system languages, developers create one app for all the app stores. One set of code in a single codebase will work on both Android and Apple OS mobile phones. Hybrid apps use coding languages as well as integrations and plugins to achieve this functionality.

If there is ever a change made to the app, instead of updating the code once for the Android app and a second time in the Apple app, developers can just do it once. Then, they can push that change to the apps on both operating systems. This saves development time, maintenance time, and money while providing a more effective platform for the user.

A hybrid app is a particular type of native app. Like native apps, users can download and install a hybrid app from app stores. However, the internal structure of native and hybrid apps is very different. Internally, hybrid apps are more like web apps. Hybrid apps lie somewhere between native and web apps.

App development

In a native app, your developers have to rewrite and redesign all the app functionality in the native development language. A hybrid app lets you write the app functionality in a single codebase. You can then wrap your code in a lightweight native app shell or container. The container enables you to take advantage of native features in your mobile devices, like hardware, calendars, and notifications.

Cost efficiency

Hybrid apps achieve the same performance and user experience as native apps at a lower cost. Your developers can build them using commonly used app development languages and technologies like JavaScript, CSS, and HTML5. They can then integrate them with hybrid app development frameworks like Ionic, Cordova, or React Native. Both time and cost of development are lower, but you can still upload them to an app store to enjoy the same reach and discoverability.

1. **React Native**

React Native, created by Facebook, is a leading choice for hybrid app development. It’s popular because it lets developers build apps using JavaScript and React, two widely known technologies. One of its standout features is its ability to produce a native look and feel on both iOS and Android. This means users get a smooth, natural experience no matter which device they use.

* React Native also includes a feature called “hot reloading.” This allows developers to see the changes they make in real-time without having to restart the entire app. This can save a lot of time and make the development process much more efficient.
* Another significant benefit is the community and ecosystem around React Native. There are many libraries, tools, and extensions available, making it easier to add complex functionalities to your app. The framework also has excellent documentation and numerous tutorials, so you won’t feel lost even if you’re just starting out.

1. **Flutter**

Flutter, created by Google, is taking the app development world by storm. Using the Dart programming language, Flutter enables developers to build natively compiled applications for mobile, web, and desktop from a single codebase. One of Flutter’s standout features is its extensive collection of customizable widgets. These widgets make it incredibly easy to design attractive and interactive user interfaces, providing a native-like experience.

* In terms of community and support, Flutter is growing rapidly. Google provides excellent documentation, and there’s a thriving community of developers who contribute plugins, extensions, and tutorials. This makes it easier for newcomers to get started and for experienced developers to find solutions to specific problems.
* Flutter also excels in cross-platform compatibility. Developers can write one set of code that works on both iOS and Android, as well as web and desktop. This significantly reduces development time and costs, making it an attractive option for businesses looking to deploy their apps across multiple platforms.

1. **Ionic**

Ionic is an open-source framework that makes hybrid app development a breeze. It’s built on web technologies like HTML, CSS, and JavaScript, making it accessible for many developers. Ionic also integrates smoothly with popular frameworks like Angular, React, and Vue. This flexibility means that you can leverage your existing skills and knowledge, making the development process quicker and more efficient. For instance, if you’re already familiar with Angular, you can use it alongside Ionic to build robust applications with minimal learning curve.

* Another fantastic aspect of Ionic is its active community. There are countless plugins and extensions available, which can add a wide range of functionalities to your app, from social media integration to analytics. The community also provides extensive documentation and tutorials, so you’ll never feel lost, even if you’re new to hybrid app development.

Program code:

package com.delaroystudios.studentmgt;

import java.security.PublicKey;

import android.os.Bundle;

import android.app.Activity;

import android.app.AlertDialog.Builder;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.view.Menu;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import com.delaroystudios.studentmgt.R;

public class StudentMainActivity extends Activity {

EditText ename,eroll\_no,emarks;

Button add,view,viewall,Show1,delete,modify;

SQLiteDatabase db;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_student\_main);

ename=(EditText)findViewById(R.id.name);

eroll\_no=(EditText)findViewById(R.id.roll\_no);

emarks=(EditText)findViewById(R.id.marks);

add=(Button)findViewById(R.id.addbtn);

view=(Button)findViewById(R.id.viewbtn);

viewall=(Button)findViewById(R.id.viewallbtn);

delete=(Button)findViewById(R.id.deletebtn);

Show1=(Button)findViewById(R.id.showbtn);

modify=(Button)findViewById(R.id.modifybtn);

db=openOrCreateDatabase("Student\_manage",

Context.MODE\_PRIVATE, null);

db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno

INTEGER,name VARCHAR,marks INTEGER);");

add.setOnClickListener(new OnClickListener()

{

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

if(eroll\_no.getText().toString().trim().length()==0||

ename.getText().toString().trim().length()==0||

emarks.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter all values");

return;

}

db.execSQL("INSERT INTO student

VALUES('"+eroll\_no.getText()+"','"+ename.getText()+

"','"+emarks.getText()+"');");

showMessage("Success", "Record added successfully");

clearText();

}

});

delete.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

if(eroll\_no.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Rollno");

return;

}

Cursor c=db.rawQuery("SELECT \* FROM student WHERE

rollno='"+eroll\_no.getText()+"'", null);

if(c.moveToFirst())

{

db.execSQL("DELETE FROM student WHERE

rollno='"+eroll\_no.getText()+"'");

showMessage("Success", "Record Deleted");

}

else

{

showMessage("Error", "Invalid Rollno");

}

clearText();

}

});

modify.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

if(eroll\_no.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Rollno");

return;

}

Cursor c=db.rawQuery("SELECT \* FROM student WHERE

rollno='"+eroll\_no.getText()+"'", null);

if(c.moveToFirst())

{

db.execSQL("UPDATE student SET

name='"+ename.getText()+"',marks='"+emarks.getText()+

"' WHERE rollno='"+eroll\_no.getText()+"'");

showMessage("Success", "Record Modified");

}

else

{

showMessage("Error", "Invalid Rollno");

}

clearText();

}

});

view.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

if(eroll\_no.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Rollno");

return;

}

Cursor c=db.rawQuery("SELECT \* FROM student WHERE

rollno='"+eroll\_no.getText()+"'", null);

if(c.moveToFirst())

{

ename.setText(c.getString(1));

emarks.setText(c.getString(2));

}

else

{

showMessage("Error", "Invalid Rollno");

clearText();

}

}

});

viewall.setOnClickListener(new OnClickListener() {

@Override

public void showMessage(String title,String message)

{

Builder builder=new Builder(this);

builder.setCancelable(true);

builder.setTitle(title);

builder.setMessage(message);

builder.show();

}

public void clearText()

{

eroll\_no.setText("");

ename.setText("");

emarks.setText("");

eroll\_no.requestFocus();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.student\_main, menu);

return true;

}

}

Output:

Write Program 6 Output