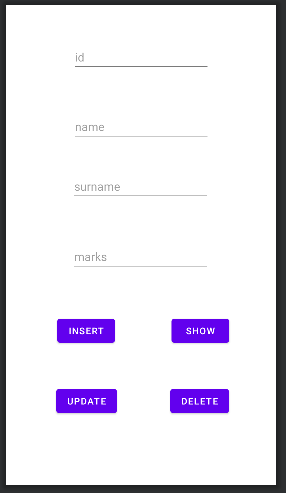
**PRACTICAL-9**

**Aim:** Create an application to make Insert, update, Delete and retrieve operation on the SQLite database.

**Design:**

****

**Code:**

MainActivity.java

package com.example.sqlite\_database;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import androidx.constraintlayout.widget.ConstraintLayout;

import android.database.Cursor;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

helper db;

EditText editText,editText1,editText2,editText3;

Button button,button1,button2,button3;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

db = new helper(MainActivity.this);

editText = findViewById(R.id.id);

editText1 = findViewById(R.id.name);

editText2 = findViewById(R.id.sn);

editText3 = findViewById(R.id.mark);

button = findViewById(R.id.bt);

button1 = findViewById(R.id.bt1);

button2 = findViewById(R.id.bt2);

button3 = findViewById(R.id.bt3);

insert();

show();

update();

delete();

}

public void insert(){

button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

boolean result=db.insertdata(editText1.getText().toString(),editText2.getText().toString(),editText3.getText().toString());

if(result == true)

Toast.makeText(MainActivity.this,"data inserted",Toast.LENGTH\_LONG).show();

else

Toast.makeText(MainActivity.this,"data not inserted",Toast.LENGTH\_LONG).show();

}

});

}

public void show(){

button1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Cursor gett = db.showdata();

if(gett.getCount() == 0){

showmessage("Error","data not found");

return; //exit from loop

}

StringBuffer buffer=new StringBuffer();

while (gett.moveToNext())

{

buffer.append("id: "+gett.getString(0)+"\n");

buffer.append("name: "+gett.getString(1)+"\n");

buffer.append("surname: "+gett.getString(2)+"\n");

buffer.append("mark: "+gett.getString(3)+"\n\n");

}

showmessage("Data",buffer.toString());

}

});

}

public void showmessage(String title,String message)

{

AlertDialog.Builder builder=new AlertDialog.Builder(MainActivity.this);

builder.setTitle(title);

builder.setMessage(message);

builder.show();

}

public void update(){

button2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

boolean

isUpdated=db.updatedata(editText.getText().toString(),editText1.getText().toString(),editText2.getText().toString(),editText3.getText().toString());

if(isUpdated==true)

Toast.makeText(MainActivity.this,"data updated",Toast.LENGTH\_LONG).show();

else

Toast.makeText(MainActivity.this,"data not updated",Toast.LENGTH\_LONG).show();

}

});

}

public void delete(){

button3.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Integer row=db.deletedata(editText.getText().toString());

if(row > 0)

Toast.makeText(MainActivity.this,"data deleted",Toast.LENGTH\_LONG).show();

else

Toast.makeText(MainActivity.this,"data not deleted",Toast.LENGTH\_LONG).show();

}

});

}

}

Helper.java

package com.example.sqlite\_database;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class helper extends SQLiteOpenHelper {

String tname = "fivece";

String col1 = "id";

String col2 = "name";

String col3 = "surname";

String col4 = "mark";

public helper(@Nullable Context context) {

super(context, "fivece.db", null, 1);

SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();

}

@Override

public void onCreate(SQLiteDatabase sqLiteDatabase) {

sqLiteDatabase.execSQL("create table fivece (id integer primary key,name text,surname text,mark integer)");

}

@Override

public void onUpgrade(SQLiteDatabase sqLiteDatabase, int oldVersion, int newVersion) {

sqLiteDatabase.execSQL("drop table if exists fivece");

onCreate(sqLiteDatabase);

}

public boolean insertdata(String name,String surname,String mark){

SQLiteDatabase sqLiteDatabase=this.getWritableDatabase();

ContentValues contentValues=new ContentValues();

contentValues.put(col2,name);

contentValues.put(col3,surname);

contentValues.put(col4,mark);

long res=sqLiteDatabase.insert(tname,null,contentValues);

if(res == -1)

return false;

else

return true;

}

public Cursor showdata(){

SQLiteDatabase sqLiteDatabase=this.getWritableDatabase();

Cursor get=sqLiteDatabase.rawQuery("select \* from "+ tname,null);

return get;

}

public boolean updatedata(String id,String name,String surname,String mark){

SQLiteDatabase sqLiteDatabase=this.getWritableDatabase();

ContentValues contentValues=new ContentValues();

contentValues.put(col1,id);

contentValues.put(col2,name);

contentValues.put(col3,surname);

contentValues.put(col4,mark);

sqLiteDatabase.update(tname,contentValues,"id = ?",new String[] {id});

return true;

}

public Integer deletedata(String id){

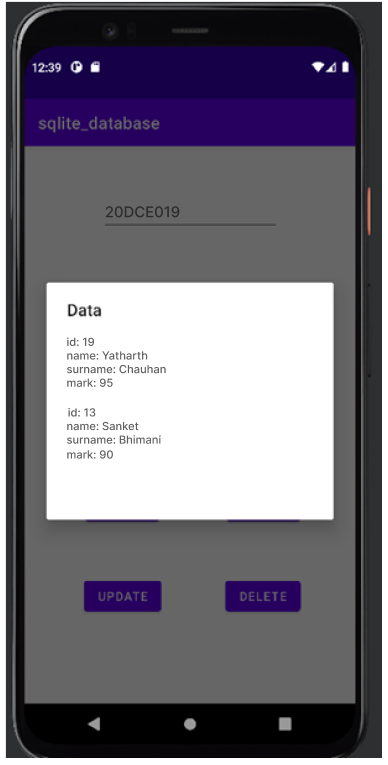
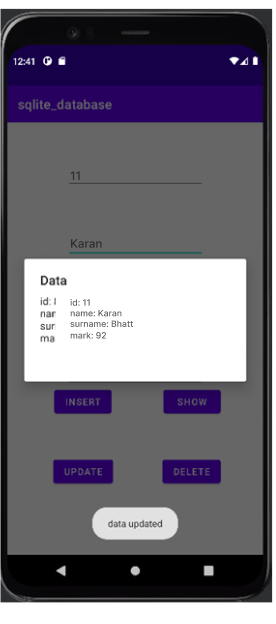
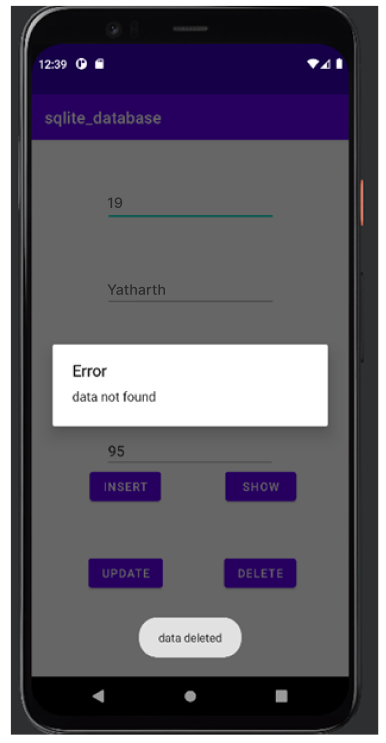
SQLiteDatabase sqLiteDatabase=this.getWritableDatabase();

return sqLiteDatabase.delete(tname,"id = ?",new String[] {id});

}

}

**Output:**

** **

**Conclusion:**

In this practical we learnt about sqlite database.

**PRACTICAL-10**

**Aim:** Introduction to I-phone & installation of x-code on MAC.

**What is Xcode?**

Xcode is an application that developers use to build apps for Apple’s various platforms such as iPhone, iPad, Macs, AppleTV and Apple Watch.

**Step – 0 Check your mac OS Version**

We will be using the latest version of Apple’s Xcode 11 to program iOS 13 apps and run our apps in Xcode’s built in iOS simulator.

Xcode is an Integrated Development Environment (IDE) developed by Apple and the vast majority of iOS developers rely on it for making iPhone or iPad applications. Xcode 11 can only be installed on a Mac running macOS 10.14.4 (Mojave) or above. But ideally, you should be running macOS 10.15.0 or above Catalina).

If you are still running an earlier OS such as Mavericks, Yosemite, El Capitan, Sierra etc, you will need to update your OS. Have a look on Apple’s website for instructions on how to do so.

Check that your OS is either Mojave 10.14.4 or above or Catalina 10.15 If your version is lower than this, head over to the Mac App Store and you can upgrade your operating system to the latest version of macOS Catalina for free.

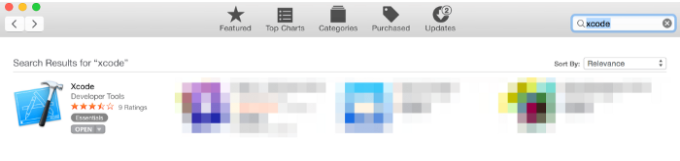
**Step 1: Open the App Store**

Once you’re sure you are running the correct version of the Mac operating system, you can get started with downloading Xcode 11 through the Mac App Store. Open the App Store app on your Mac. By default, the App Store is in the Dock. You can also find it in your Launchpad.

****

**Step 2: Search for Xcode**

In the search field in the top-right corner, type Xcode and press the Return key.



**Step 3: Install Xcode**

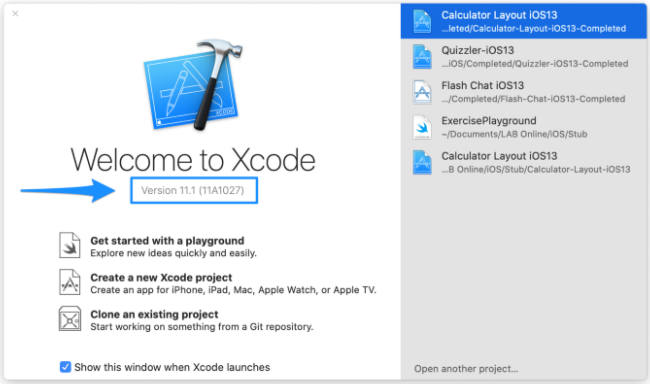
Xcode is a free application developed by Apple, so just click the “Get” or “Download” button and start the installation process.



Xcode is several gigabytes in size so downloading it could take a while. By default, Xcode is downloaded into your /Applications directory.

**Step 4: Launch Xcode**

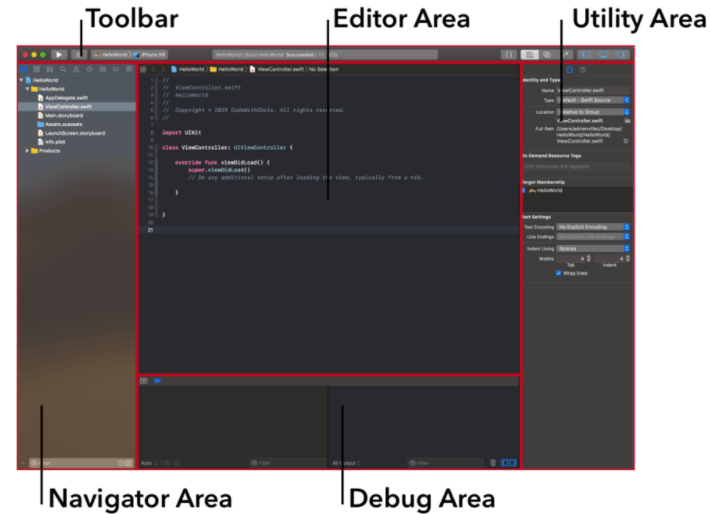
Launch Xcode. Do you see the Welcome to Xcode window and the version is 11.0 or above (e.g 11.1 or 11.1.2 etc.)? If yes, then great, we’re all done!



By the end of this chapter, you’ll have a great understanding of how to navigate the development environment and how to create your first Xcode project.

Before we dive in, keep this Apple Documentation for Xcode handy.

Let’s now look at the diagram below. If your interface looks different, make sure you have Xcode 11 and not an earlier version.

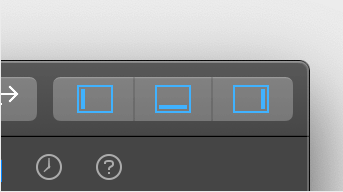


The reason this is a great diagram is because it lets me refer to these different sections of the interface and you can refer back to this diagram to see what I’m talking about!

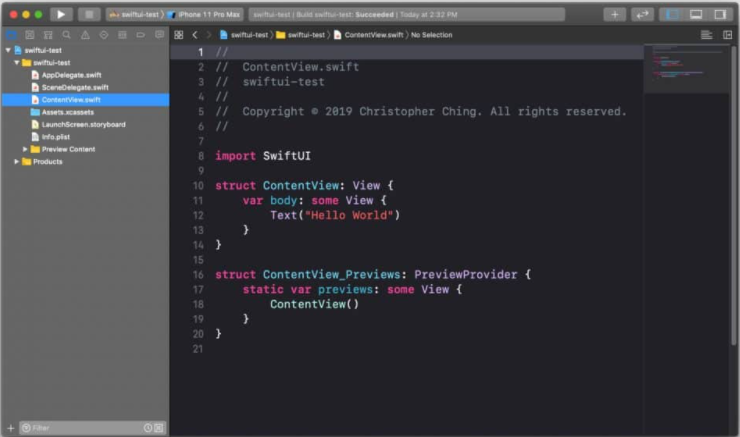
As you can see from the diagram, there are 5 major areas: the Navigator, Editor, Utility Area, Toolbar and Debug Area. We’ll cover each area in detail later but for now, let’s talk about general navigation of Xcode.

Keep in mind that you can adjust the size of each of those panes by hovering your cursor over the boundaries of each area and dragging.

You can also show and hide the various areas as needed via the “View” buttons in the upper right hand corner:



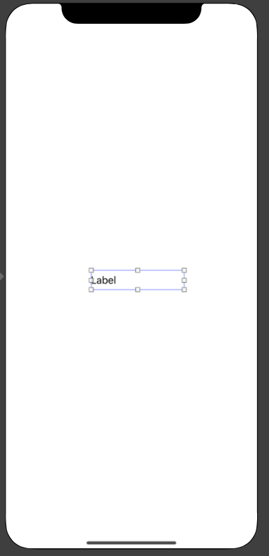
This can be helpful, for example, when you’re writing code and you don’t need the debugger area or the utility area. Then you can use the View buttons to hide those 2 panes to give your editor more visible space.



**PRACTICAL-11**

**Aim:** Create IOS application that will display “Custom Message” in the middle of the screen in the Black color with the Yellow background.

**Design:**

****

**Code:**

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var customlabel: UILabel!

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view.

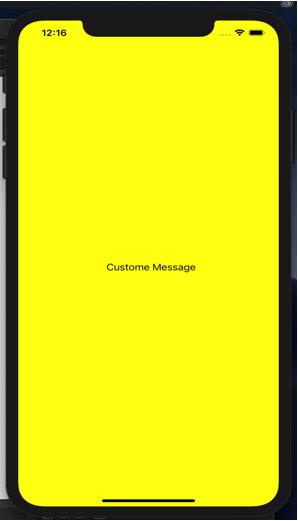
        customlabel.textColor = UIColor.black

        customlabel.text = "Custome Message"

        self.view.backgroundColor = UIColor.yellow

    } }

**Output:**

****

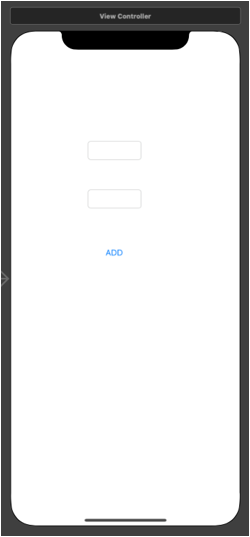
**Conclusion:**

In this practical we learnt how to set background colour.

**PRACTICAL-12.1**

**Aim:** Create an IOS application to calculate sum of two numbers and gives result in Toast Message.

**Design:**

****

**Code:**

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var Number1: UITextField!

    @IBOutlet weak var Number2: UITextField!

    @IBOutlet weak var Add: UIButton!

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view.

    }

    @IBAction func Addclick(\_ sender: Any) {

        let a:Int! = Int(Number1.text!)

        let b:Int! = Int(Number2.text!)

        let c = a + b

        //Result.text = String(c)

        Toast(String(c))

    }

}

extension ViewController

{

    func Toast(\_ message:String)

    {

        var Toastlabel = UILabel(frame: CGRect(x: self.view.frame.width/2-75, y: self.view.frame.height-100, width: 150, height: 40))

        Toastlabel.textAlignment = .center

        Toastlabel.text = message

        Toastlabel.textColor = UIColor.black

        Toastlabel.backgroundColor = UIColor.red.withAlphaComponent(0.5)

        Toastlabel.alpha = 1.0

        self.view.addSubview(Toastlabel)

        UIView.animate(withDuration: 3.0, delay: 1.0, options: .curveEaseInOut, animations: {

            Toastlabel.alpha = 0.0

        }) { (isCompleted) in

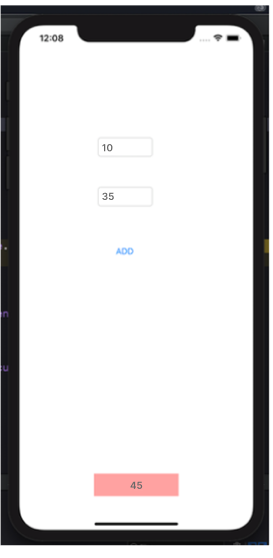
            Toastlabel.removeFromSuperview()

        }

    }

}

**Output:**



**Conclusion:**

In this practical we learnt how to calculate sum of two numbers.

**PRACTICAL-12.2**

**Aim:** Design an IOS application representing a simple calculator.

**Design:**

****

**Code:**

class ViewController: UIViewController {

    @IBOutlet weak var Number1: UITextField!

    @IBOutlet weak var Number2: UITextField!

    @IBOutlet weak var labelanswer: UILabel!

    @IBOutlet weak var btnadd: UIButton!

    @IBOutlet weak var btnsubtract: UIButton!

    @IBOutlet weak var btnmultiply: UIButton!

    @IBOutlet weak var btndivide: UIButton!

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view.

    }

    @IBAction func btnaddclick(\_ sender: Any) {

        let a:String! = Number1.text

        let b:String! = Number2.text

        let a1:Int! = Int(a)

        let a2:Int! = Int(b)

        let c = a1 + a2

        labelanswer.text = "Answer is \(c)"

        print("add")

    }

    @IBAction func btnsubtractclick(\_ sender: Any) {

        let a:String! = Number1.text

        let b:String! = Number2.text

        let a1:Int! = Int(a)

        let a2:Int! = Int(b)

        let c = a1 - a2

        labelanswer.text = "Answer is \(c)"

        print("subtract")

    }

    @IBAction func btnmultiplyclick(\_ sender: Any) {

        let a:String! = Number1.text

        let b:String! = Number2.text

        let a1:Int! = Int(a)

        let a2:Int! = Int(b)

        let c = a1 \* a2

        labelanswer.text = "Answer is \(c)"

        print("multiply")

    }

    @IBAction func btndivideclick(\_ sender: Any) {

        let a:String! = Number1.text

        let b:String! = Number2.text

        let a1:Float! = Float(a)

        let a2:Float! = Float(b)

        let c = a1 / a2

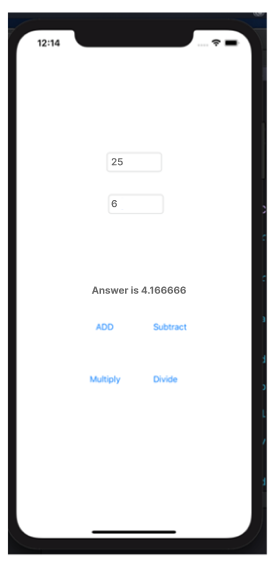
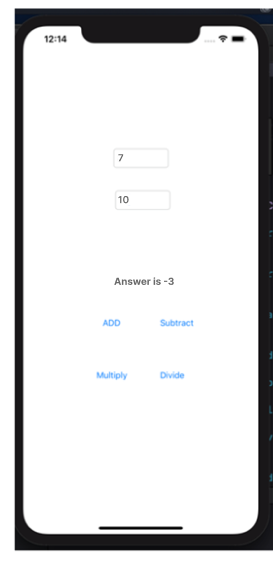
        labelanswer.text = "Answer is \(c)"

        print("divide")

    }

}

**Output:**



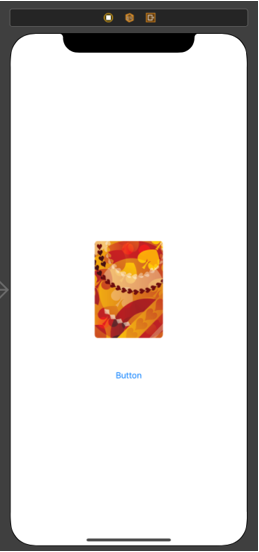
**Conclusion:**

In this practical we learnt how to construct simple calculator

**PRACTICAL-13**

**Aim:** Create an I-OS Application to Change the Image Displayed on the Screen

**Design:**

****

**Code:**

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var card: UIImageView!

    @IBOutlet weak var imgchange: UIButton!

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view.

    }

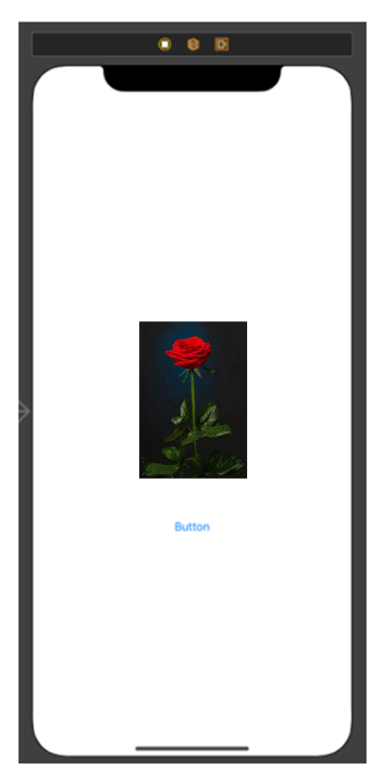
    @IBAction func imgchangeclick(\_ sender: Any) {

        card.image = UIImage(named: "flower")

    }

}

**Output:**



**Conclusion:**

In this practical we learnt how to change image displayed on the screen.

**PRACTICAL-14**

**Aim:** Understanding of UI:Create an UI such that, one screen has list of all the types of cars.On selecting of any car name, next screenshould show Car details like: name, launched date,company name, images(using gallery) if available, show different colors in which it is available.

**Design:**

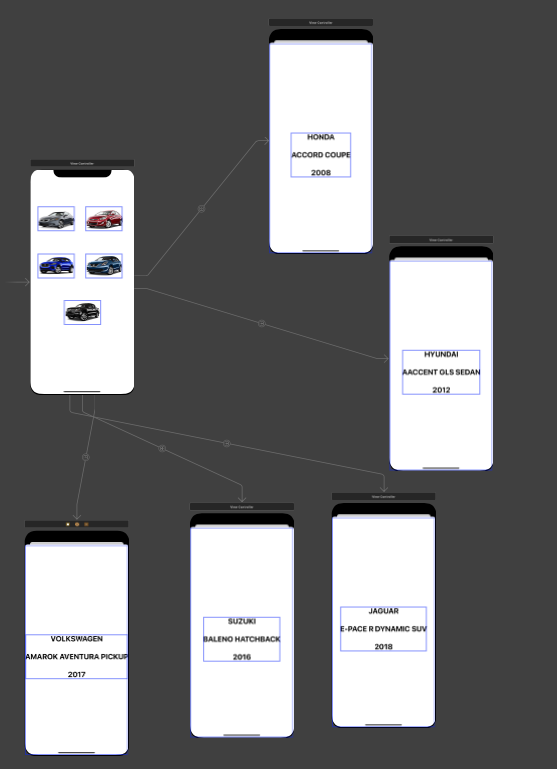
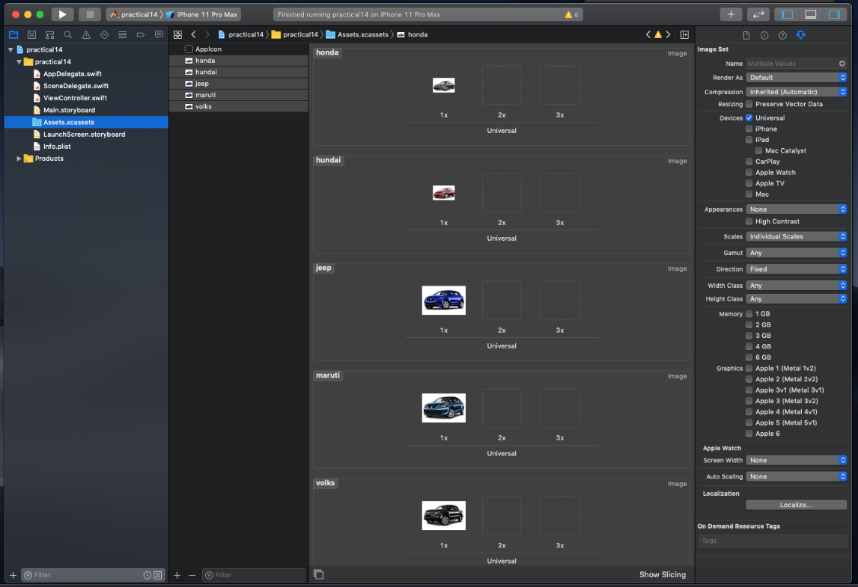
****

IMAGE ASSETS:

****

**Code:**

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {

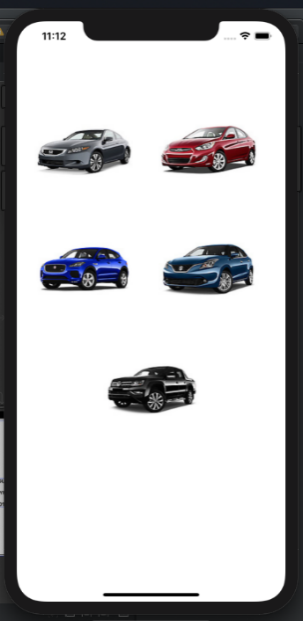
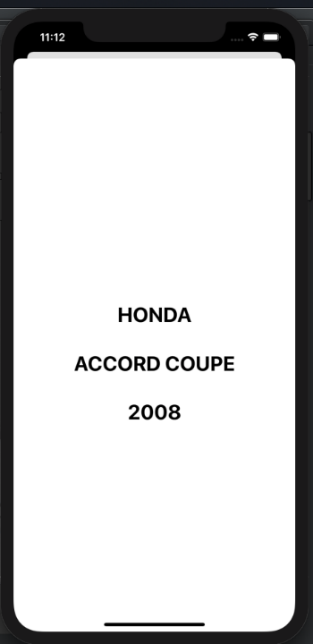
        super.viewDidLoad()

        // Do any additional setup after loading the view.

    }

}

**Output:**

** **

**Conclusion:**

In this practical we learnt how to change image by clicking on it.