MAHENDRACOLLEGEOFENGINEERING

Approved by AICTE and Affiliated by Anna University, Chennai

NAACAccredited–RecognizedU/S2(F)&12(B)ofUGCAct 1956

Salem-ChennaiHighwayNH79,Minnampalli, Salem-636106

DEPARTMENTOFINFORMATIONTECHNOLOGY

IT3681

MOBILE APPLICATION DEVELOPMENT LABORATORY

RECORD NOTE BOOK

CLASS:III YEAR/ VI SEMESTER

MAHENDRA

COLLEGEOF ENGINEERING

NAACAccredited

ApprovedbyAICTEandAffiliatedbyAnnaUniversity,Chennai Salem-

Chennai Highway NH 79, Minnampalli, Salem-636106

Department of

_____

LABORATORY RECORD

Certified to be the Bonafide of workdone by

Name: _____Register No: _____

Class: _____Branch: _____

Laboratory Name: _____

HEAD OF THE DEPARTMENT STAFF-INCHARGE
DATE:

Submitted for the University Practical Examinationon

_____

INTERNAL EXAMINER EXTERNAL EXAMINER

S.NO	DATE	TITLE	PAGENO	MARK	SIGN

INSTITUTIONVISIONANDMISSION

VISION

Mahendra College of Engineering is committed to be a leader in Higher Education achieving excellence through world class learning environment for Science and Technology with a blend of advanced research to create ethical and competent professionals

MISSION

- To provide a conductive atmosphere to impart innovative knowledge and commendable skills through quality education by continuous improvement and customization of teaching.
- To nurture research attitude and bring about tangible developments with dynamic Industry - Institute Interaction.
- To create society oriented citizens with professional ethics.

DEPARTMENT VISIONANDMISSION

VISION

Tobecomeadepartment, producinggraduateswithgoodtechnicalskills inemergingareasof Information Technology, through value based education and research.

MISSION

- Toprovideexposuretostudentstotheemergingtechnologies inHardwareandSoftware.
- Toinculcatestudentswithsound applicationknowledge.
- ToestablishstrongIndustry-InstituteInteraction.

PROGRAMME SPECIFIC OUTCOMES(PSOs)

Toensure graduates

- Haveproficiencyinprogrammingskillstodesign, developandapplyappropriatetechniques, tosolve complex engineering problems.
- Haveknowledgetobuild,automateand managebusinesssolutionsusingcuttingedgetechnologies.
- Haveexcitementtowards researchinappliedcomputertechnologies.

PROGRAMOUTCOMES(POs)

1. Engineeringknowledge:

Applytheknowledgeofmathematics,science,engineeringfundamentals,andanengineering specialization to the solution of complex engineering problems.

2. **Problemanalysis:**

Identify,formulate,reviewresearchliterature,andanalyzecomplexengineeringproblemsreachingsubstantiatedconclusionsusingfirstprinciplesofmathematics,naturalsciences, and engineering sciences.

3. **Design/developmentofsolutions:**

Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conductinvestigationsofcomplexproblems:**

Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Moderntool usages:**

Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including predictionand modeling to complex engineering activities withanunderstanding ofthe limitations.

6. **Theengineerandsociety:**

Applyreasoning informed bythecontextualknowledgeto assesssocietal, health,safety, legaland cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environmentandsustainability:**

Understandthe impact oftheprofessionalengineering solutions insocietalandenvironmentalcontexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:**

Applyethicalprinciples andcommit toprofessionalethicsandresponsibilitiesandnormsofthe engineering practice.

9. **Individualandteamwork:**

Functioneffectivelyasanindividual, andasa member orleader indiverseteams, and in multidisciplinary settings.

10. **Communications:**

Communicate effectively on complex engineering activities with the engineering communityand with society at large, such as, being able to comprehend and write effective reports and design documentation,make effective presentations, and give and receive clear instructions.

11. **Projectmanagementandfinance:**

Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-longlearning:**

Recognize the need for, and have the preparation and ability to engage in independent and life- long learning in the broadest context of technological change.

MOBILE APPLICATIONS DEVELOPMENT LABORATORY

COURSEOBJECTIVES:

- The objective of this course is to enable the students to
- Use Flutter/Kotlin multi-platform environment for building cross-platform mobile applications.
- Demonstrate the knowledge of different programming techniques and patterns for mobile application development.
- Identify the components and structure of mobile application development frameworks.
- Understand the capabilities and limitations of different platforms.
- Design and develop real-time mobile applications.

LIST OF EXPERIMENTS:

1. Study and installation of Flutter/Kotlin multi-platform environment
2. Develop an application that uses Widgets, GUI components, Fonts, and Colors.
3. Develop a native calculator application.
4. Develop a gaming application that uses 2-D animations and gestures.
5. Develop a movie rating application (similar to IMDB)
6. Develop an application to connect to a web service and to retrieve data with HTTP.
7. Develop a simple shopping application.
8. Design a web server supporting push notifications
9. Develop an application by integrating Google maps
10. Mini Projects involving Flutter/Kotlin multi-platform

TOTAL : 45 PERIODS

COURSE OUTCOMES:
On successful completion of this course, the student should be able to
CO1:Design and build simple mobile applications supporting multiple platforms
CO2:Apply various programming techniques and patterns to build mobile applications.
CO3:Build real-time mobile applications for society/environment
CO4:Build gaming and multimedia based mobile applications
CO5:Build AI based mobile applications for society/environment following ethical practices

CO's-PO's & PSO's MAPPING

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	3	3	3	1	3	1	1	1	2	1	1	1	2	2	2
2	3	3	3	2	3	1	1	1	2	1	1	1	2	2	2
3	3	3	3	3	3	3	2	2	3	3	3	3	3	3	3
4	3	3	3	3	3	2	1	1	1	1	2	1	1	2	2
5	3	3	3	3	2	1	1	1	1	1	1	1	2	2	2
Avg.	3	3	3	3	2	1	1	1	1	1	1	1	2	2	2

1 - low, 2 - medium, 3 - high, '-"- no correlation

ExNo: 1	Study and installation of Flutter/Kotlin multi - platform environment
Date:	

Aim:

To Study and install Flutter/Kotlin multi-platform environment

Procedure

: Flutter:

InstallFlutter SDK:

Download the Flutter SDK from the official Flutter website.
Extract the downloaded archive and add the flutter/bin directory to your system PATH.

SetupanIDE(IntegratedDevelopmentEnvironment):

You can use Android Studio,Intell IDEA,orVisual Studio Code for Flutter development.
Install the Flutter and Dart plugins for your chosen IDE.

Run flutter doctor:

Open a terminal and run flutter doctor to check for any dependencies that need to be installed.Follow the instructions provided by flutter doctor to resolve any issues.

Create a Flutter Project:

Run the following commands in your terminal:

Flutter
createmy_flutter_project cd
my_flutter_project

RuntheApp:
Connect a device or start an emulator and run flutter run in the project directory.

Kotlin:

Install Kotlin:
You can install Kotlin by following the instructions on the official Kotlin website.
Setup anIDE:
Kotlin Multiplatform projects are well-supported in Intell iJIDEA. Install the Kotlin plugin for your IDE.

Createa Kotlin MultiplatformProject:

Create an ewKotlin Multiplatform project using a template or set it up manually.

Configure Shared Code:

Define the common code that will be shared a crossplatforms.This code will be written in Kotlin and can include business logic,datamodels,etc.

Platform-SpecificCode:

Implement platform-specificcodeforAndroidandiOS. This involves creating separate modules or directories for each platform and writing thenecessary code in Kotlin(for commonlogic) and Swift/Kotlin(forplatform-specificcode).

Buildand Run:

Buildthe project usingthe appropriate build command for your project tstructure (e.g., Gradle forAndroid). Run the app on Android and iOSdevices/emulators

Result:

Thus the flutter application was installed and verified successfully.

ExNo:2	Application usingWidgets, GUIcomponents,Fonts,andColors.
Date:	

Aim:

Develop an application that uses Widgets, GUIcomponents ,Fonts,and Colors.

Procedure:

1) Open your preferred Kotlin IDE.
2) Create a new Kotlin project.
3) Download the Java FXSDK from the official website.
4) Extract the downloaded archive to a location on your computer.
5) In your Kotlin project,add the JavaFX library to your project's dependencies.
6) Create a new Kotlin file for your main application class,e.g.,**MyApp.kt.**
7) Copy and paste the Kotlin code provided in the previous response in to this file.
8) Open the Run/Debug Configuration in your IDE.
9) Add the following VM options:
   --module-path/path/to/javafx-sdk-17/lib--add-modulesjavafx.controls,javafx.fxml
   Replace /path/to/javafx-sdk-17withthe actualpathtoyourJavaFX SDK.
   Run the main function in your **MyApp.kt** file.

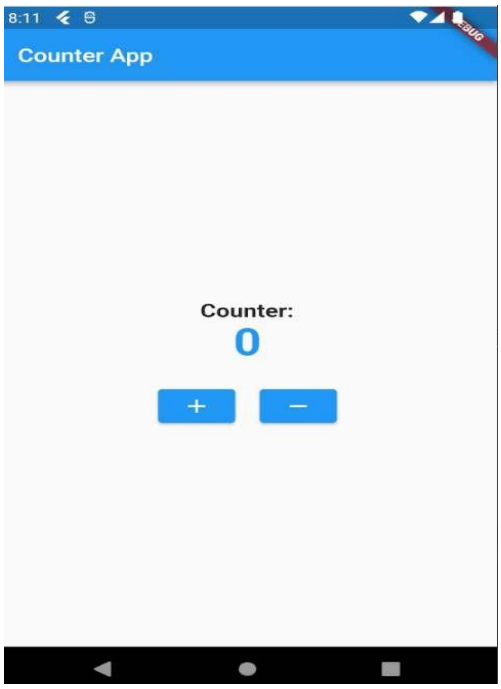
Program:

```
importjavafx.application.Applicationi

mportjavafx.geometry.Insets

importjavafx.scene.Scene

importjavafx.scene.control.Buttonimp

ortjavafx.scene.control.ComboBoximp

ortjavafx.scene.control.Labelimportjav

afx.scene.control.TextFieldimportjavaf

x.scene.layout.VBox

importjavafx.scene.text.Font
importjavafx.stage.Stage
```

```
classMyApp:Application(){
  override funstart(primaryStage:Stage){

    primaryStage.title="WidgetApp"

    valroot=VBox(10.0)

    root.padding=Insets(10.0)

    //Set a customfont

    valcustomFont=Font("Arial", 12.0)

    //Set backgroundcolor

    root.style="-fx-background-color:#f0f0f0;"

    //Createalabel

    vallabel=Label("Welcome toWidgetApp")

    label.font

    =customFontroot.children.add(label)

    //Createa button

    valbutton=Button("ClickMe!")

  button.font=customFontbutton.setOnA

    ction{onButtonClick()}root.children.ad

    d(button)

    //Createatextfield

    valtextField=TextField()

    textField.font=customFontr

    oot.children.add(textField)

    //Createa combobox

    valcomboBoxValues=listOf("Option1","Option2","Option 3")

    valcomboBox =ComboBox<String>()

    comboBox.items.addAll(comboBoxValues)
```

```
    comboBox.font
        =customFontroot.children.ad
        d(comboBox)
        valscene=Scene(root,300.0,200.0)
        primaryStage.scene=scene
        primaryStage.show()
    }
    privatefunonButtonClick(){
        println("ButtonClicked!")
    }
}
funmain(){
    Application.launch(MyApp::class.java)
}
```

Sample output



RESULT:

Thus the application using Widgets, GUI components,Fonts ,and Colors was executed successfully.

ExNo: 3	Develop a native calculator application.
Date:	

Aim:

To develop a native calculator application.

Procedure:

CreateanewAndroid StudioProject:

Open Android Studio and create a new project.

Choose"Empty Activity" template.

Design theUI:

Open res/layout/activity_main.xml and replace its content with the following Xmlcode

Define ButtonIDs andOperators:

Open Main Activity. Kt and replace its content with the following kotlin code

Run yourapp:

Connect your Android device or start an emulator.

Click on the"Run" button in Android Studio.

Program:

Xml:

```
<?xmlversion="1.0"encoding="utf-8"?>
<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"android:layout_width="mat
   ch_parent" android:layout_height="match_parent"
   tools:context=".MainActivity">
   <EditText

   android:id="@+id/inputText"
  android:layout_width="match_parent"
   android:layout_height="wrap_conten
```

```
t" android:layout_marginTop="16dp"

android:layout_marginBottom="16dp"

android:layout_alignParentTop="true"

android:layout_centerHorizontal="true

" android:hint="Enterexpression"

android:inputType="none"

android:textAlignment="center"/>
```

```
    <GridLayout
        android:id="@+id/gridLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/inputText"
        android:layout_marginTop="16dp"
        android:columnCount="4"
        android:orientation="horizontal">
        <!--Buttonsfordigitsandoperatorsgohere-->
    </GridLayout>
</RelativeLayout>
```

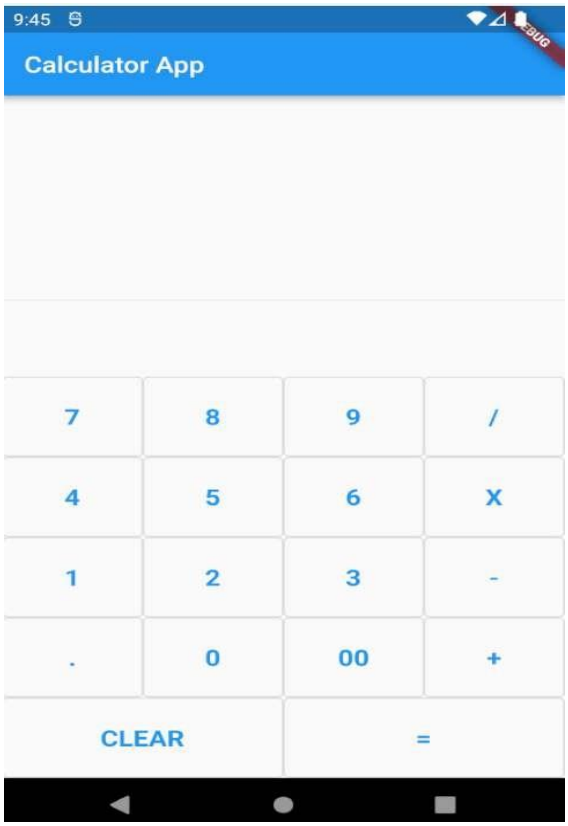
Kotlin:

```
importandroid.os.Bundleimport
android.view.Viewimportandro
id.widget.Buttonimportandroid
.widget.EditText
importandroidx.appcompat.app.AppCompatActivityi
mportnet.objecthunter.exp4j.ExpressionBuilder
classMainActivity :AppCompatActivity(){
    privatelateinitvarinputText:EditText

    overridefunonCreate(savedInstanceState:Bundle?){su
        per.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        inputText=findViewById(R.id.inputText)
        valbuttons=intArrayOf(
            R.id.btn0,R.id.btn1,R.id.btn2,R.id.btn3,
```

```
            R.id.btn4,R.id.btn5,R.id.btn6,R.id.btn7,
            R.id.btn8,R.id.btn9,R.id.btnAdd,R.id.btnSub,
            R.id.btnMul,R.id.btnDiv,R.id.btnDot,R.id.btnEquals
        )
        for(buttonIdinbuttons){
            valbutton=findViewById<Button>(buttonId)
            button.setOnClickListener{onButtonClick(it)}
        }
    }
  privatefunonButtonClick(view: View){
        valcurrentText=inputText.text.toString()
        valbuttonText=(viewasButton).text.toString()
        when(view.id) {
            R.id.btnEquals->evaluateExpression(currentText)
            R.id.btnDot->handleDotButtonClick(currentText)
            else->inputText.setText("$currentText$buttonText")
        }
  }
    privatefunevaluateExpression(expression:String){
        try {
            valresult=ExpressionBuilder(expression).build().evaluate()
            inputText.setText(result.toString())

} catch(e:Exception){
            inputText.setText("Error")
        }
    }privatefunhandleDotButtonClick(currentText:String){
        vallastOperatorIndex=getLastOperatorIndex(currentText)
```

```
        vallastDotIndex=currentText.lastIndexOf(".")
        if(lastDotIndex>lastOperatorIndex){
            //Ifthereisalreadyadotafterthelastoperator,donothing.
            return
        }inputText.setText("$currentText.")
    }privatefungetLastOperatorIndex(text:String):Int {
        valoperators=charArrayOf('+','-','*','/')
        varlastOperatorIndex=-1
        for(operatorinoperators){
            valindex=text.lastIndexOf(operator)
            if(index>lastOperatorIndex) {
                lastOperatorIndex=index
            }
        }returnlastOperatorIndex
    }
}
```

Sample output



RESULT:

Thus the Calculator application was executed successfully.

ExNo: 4	Gaming Application using2-D animations and gestures.
Date:	

Aim:

To develop a gaming application that uses2-D animations and gestures.

Procedure:

1) Install Android Studio.
2) Setup a new project using Kotlin and integrate LibGDX.
3) Create2D sprites for characters,backgrounds,and other elements using graphic design tools.
4) Animate characters using LibGDX's animation support.
5) Utilize LibGDX's gesture detection or implement a custom gesture recognition system.
6) Implement controls that respond to gestures for player navigation and interaction.
7) Define core game mechanics,challenges,and rewards,integrating gestures into gameplay.
8) Create different levels with increasing difficulty and design environments that encourage gesture-based interactions.
9) Add background music,sound effects,and any necessary audio assets.
10) Regularly test the game on Android devices to identify and fix bugs.Ensure smooth gesture-based interactions.
11) Optimize the game for performance, considering the limitations of mobile devices.
12) Prepare the game for release by configuring Android-specific settings.
13) Publish the game on the Google PlayStore.

Program:

Main GameClass:

```
import

com.badlogic.gdx.ApplicationAdapterimport

com.badlogic.gdx.Gdx

importcom.badlogic.gdx.graphics.GL20

importcom.badlogic.gdx.graphics.Texture

importcom.badlogic.gdx.graphics.g2d.SpriteBatch

classGestureQuest:ApplicationAdapter(){

privatelateinitvarbatch:SpriteBatch
```

```
    private lateinitvarimg:Texture

overridefuncreate(){
        batch=SpriteBatch()
        img=Texture("badlogic.jpg") //Replacewithyourown image file
    }overridefunrender(){Gdx.gl.glClearColor(1f,1f,1
        f,1f)
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT)
        batch.begin()
        batch.draw(img,0f,0f)
        batch.end()
    }overridefundispose(){
        batch.dispose()
        img.dispose()
    }
}
```

GestureRecognition:

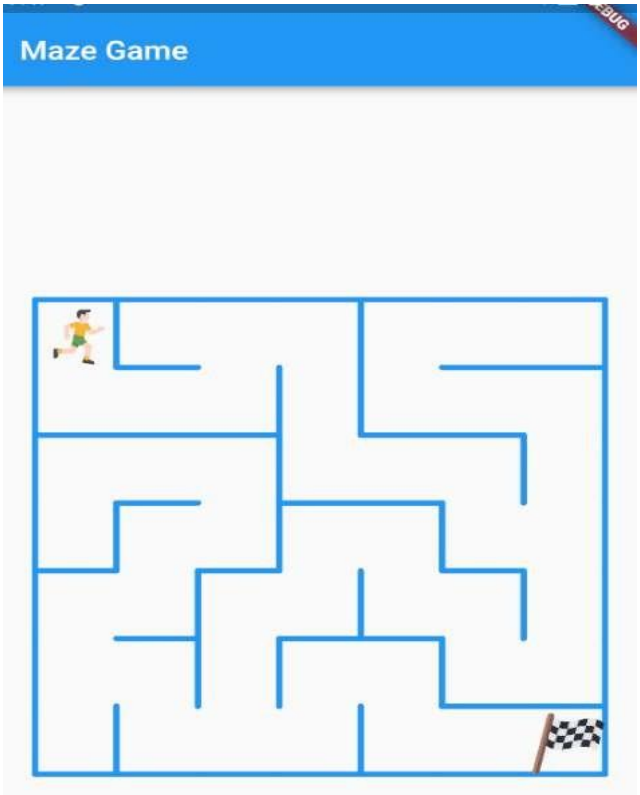
```
importcom.badlogic.gdx.Gdx
importcom.badlogic.gdx.InputProcessorcla
ssMyInputProcessor:InputProcessor{
    override funtouchDown(screenX:Int,screenY:Int,pointer:Int,button:Int):Boolean{
        //Handletouchdowneventret
        urntrue
    }
    override funkeyDown(keycode: Int):Boolean{
        //Handlekeydowneventret
        urnfalse
```

```
//ImplementotherInputProcessormethodsasneeded
   override funtouchUp(screenX:Int,screenY:Int,pointer:Int,button:Int){}
   override funkeyUp(keycode:Int){}
   override funkeyTyped(character:Char)=false
   override funtouchDragged(screenX:Int,screenY:Int,pointer:Int){}
   override funmouseMoved(screenX: Int,screenY:Int){}
   override funscrolled(amount:Int){}
}
```

To usethisinputprocessor in your maingameclass,addthefollowingcodein the `create`method:

```
valinputProcessor=MyInputProcessor()
Gdx.input.inputProcessor=inputProcessor
```

Sample output



RESULT:

Thus the Gaming application was executed successfully.

ExNo: 5	Movieratingapplication(similartoIMDB)
Date:	

Aim:

To develop a movie rating application(similartoIMDB).

Procedure:

1) Install Java and Kotlin.
2) Setup you rpreferred IDE(Intell iJIDEA is popular for Kotlin development).
3) Use the Spring Initializer to generate a new Spring Boot project with Kotlin and your preferred dependencies(SpringWeb,SpringDataJPA,etc.).
4) Define entities for movies,users,reviews,etc.
5) Use Spring Data JPA to interact with the database.
6) Create controllers for handling HTTP requests(e.g.,movielisting,userauthentication, review submission).
7) Implement services to handle business logic.
8) Setup Spring Security for user authentication.
9) Integrate Spring Security for user registration,login,and profile management.
10)Allow users to submit and retrieve movie ratings.
11)Calculate and display average ratings.
12)Implement end points for users to submit and retrieve reviews.
13)Add a comment system for reviews.
14)Use a front-end framework (e.g.,React,Angular,Vue)ortemplatingengine(Thymeleaf) To create the user interface.
15)Implement pages for browsing movies,viewing details,and submitting reviews.
16)Implement search functionality using query parameters or a dedicated search end point.
17) Write unit tests and integration tests for your Kotlin code.
18)Use testing libraries like JUnit and Mockito.
19)Deploy your Spring Boot application to a platform like Heroku, AWS, or any other of your choice.
20)Implement security best practices,such as input validation and securing APIend points.
21)Gather user feedback andc onsider adding new features or improving existing ones.
22)Implement monitoring tools to track application performance.
23)Regularly update dependencies and address security vulnerabilities.

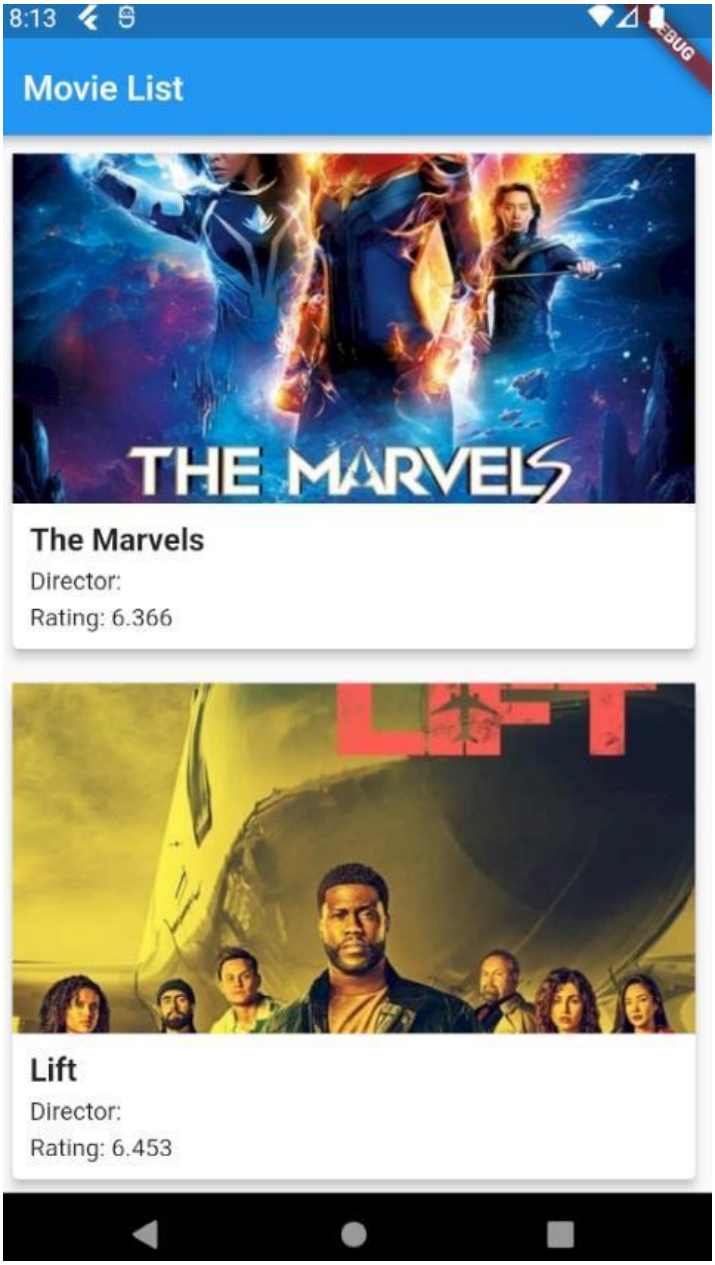
Program:

```
data class Movie(valtitle:String,valgenre:String,valreleaseYear:Int)

dataclassRating(valuser:String,valmovie:String,valrating:Double)

classMovieRatingSystem{
```

```
    privatevalmovies=mutableListOf<Movie>()

privatevalratings =mutableListOf<Rating>()
    funaddMovie(movie:Movie){
        movies.add(movie)
    }
    funrateMovie(user:String,movieTitle:String,rating:Double){
        valmovie=movies.find{it.title==movieTitle }
        if(movie !=null){
            valuserRating=Rating(user,movieTitle, rating)
            ratings.add(userRating)
            println("Ratingaddedsuccessfully.")
        } else{
            println("Movienotfound.")
        }
    }
  fungetAverageRating(movieTitle:String):Double{
        valmovieRatings=ratings.filter{ it.movie==movieTitle }
        returnif(movieRatings.isNotEmpty()){
            valtotalRating=movieRatings.map{ it.rating }.sum()
            totalRating/movieRatings.size
        }else{
            0.0
        }
    }
    funlistMovies(){
        movies.forEach{
            println("${it.title}(${it.releaseYear})-${it.genre}")
```

```
}
   }
}
funmain(){
   valmovieRatingSystem=MovieRatingSystem()
   //AddingmoviesmovieRatingSystem.addMovie(Movie("Inception","Sci-Fi",2010))
   movieRatingSystem.addMovie(Movie("TheShawshankRedemption","Drama",1994))
   movieRatingSystem.addMovie(Movie("TheDark Knight","Action",2008))
   //Listingmoviesprintln("Ava
   ilableMovies:")
movieRatingSystem.listMovies()
   //Rating  movies
   movieRatingSystem.rateMovie("user1","Inception",9.0)
   movieRatingSystem.rateMovie("user2","Inception",8.5)
   movieRatingSystem.rateMovie("user1","TheShawshankRedemption",9.5)
   //Gettingaverageratings
   println("AverageRatingforInception: ${movieRatingSystem.getAverageRating("Inception")}")
   println("AverageRatingforTheShawshankRedemption:
${movieRatingSystem.getAverageRating("TheShawshankRedemption")}")
   println("AverageRatingforTheDarkKnight:${movieRatingSystem.getAverageRating("The
DarkKnight")}")
}
```

Sample output



RESULT:

Thusthe Movie Rating Application was executed and verified successfully.

ExNo:6	Application to connect to a web services and to retrieve data with HTTP.
Date:	

Aim:

To develop an application to connect to a webservice and to retrieve data with HTTP.

Procedure:

1) Use your preferred IDE(suchasIntelliJIDEA)orabuildtool(suchasGradle)to create a new Kotlin project.
2) If you're using Gradle,add the following dependencies to your build.gradle.kts (for KotlinDSL) or build.gradle(forGroovyDSL):
   implementation"io.ktor:ktor-client-apache:$ktor_version"
   implementation"io.ktor:ktor-client-core:$ktor_version"
   implementation"io.ktor:ktor-client-serialization:$ktor_version"
3) Make sure to replace$ktor_version with the version of Kt or you want to use.
4) In your project,create a Kotlin file(e.g.,Main.kt)to write your application code.
5) Write the code in'Main.kt'
6) Use your IDE or build to ol to run the Kotlin application. The program will make a GET Request to the specified API URL,retrieve data,and print it to the console.

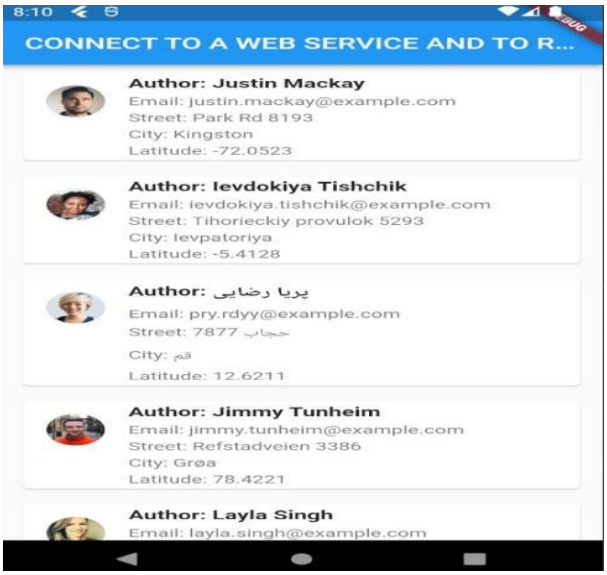
Program:

Main.kt

```
import io.ktor.client.HttpClient

importio.ktor.client.engine.apache.Apachei

mportio.ktor.client.request.get

suspendfunfetchData(url:String):String{

   valclient =HttpClient(Apache)

   returntry{
```

```
        valresponse=client.get<String>(url)
        response
    } finally {
        client.close()
    }

}
funmain(){
    valapiUrl="https://api.example.com/data"
    try {
        valresult =fetchData(apiUrl)
        println("Datareceived:$result")
    } catch(e:Exception){
        println("Error:${e.message}")
    }
}
```

Sample output



RESULT:

Thus the web service application was created and data is retrieved successfully

ExNo:7	Development of simple shopping application.
Date:	

Aim:

To Develop a simple shopping application.

Procedure:

1) Define the Product Data Class
2) Define Procedures for Displaying Products,Adding to Cart,and ViewingCart
3) Implement the MainShopping CartLogic
4) Compile and run the Kotlin program
5) interact with the simple shopping app by choosing option sfrom the menu.

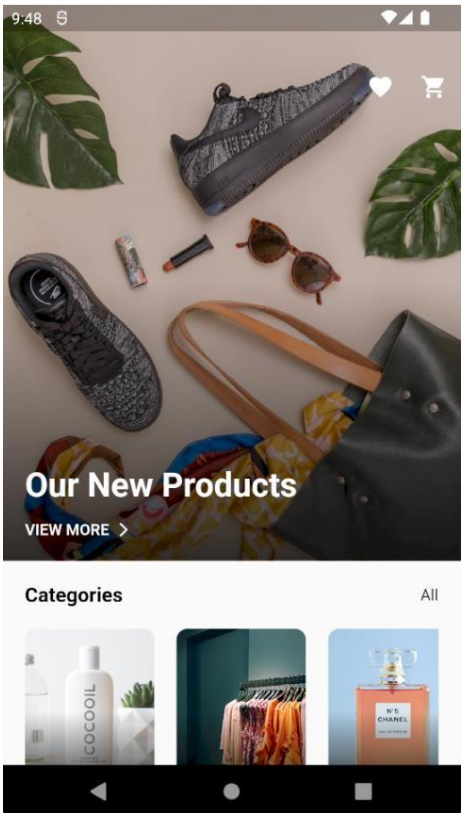
Program:

```
import java.util.Scanner

data class Product(valid:Int,valname:String,valprice:Double)

fundisplayProducts(products:List<Product>){

   println("\nAvailableProducts:")

   for(product inproducts){

      println("${product.id}.${product.name}-$${product.price}")

   }

}

funaddToCart(cart:MutableList<Product>,product:Product){

   cart.add(product)

   println("${product.name}addedtothe cart.")

}
funviewCart(cart:List<Product>){
```

```
if(cart.isEmpty()) {
        println("Yourcartisempty.")
    } else {


println("YourCart:")
        for((index,item)incart.withIndex()){
            println("${index+1}.${item.name}-$${item.price}")
        }
        valtotalPrice =cart.sumByDouble{ it.price }
        println("Total:$$totalPrice")
    }
}
funmain(){
    valscanner=Scanner(System.`in`)
    valcart=mutableListOf<Product>()
    valproducts = listOf(
        Product(1,"Product    A",10.99),
        Product(2,  "ProductB",  19.99),
        Product(3,"ProductC",7.49)
    )
    println("WelcometotheSimpleShoppingApp!")
    while(true){
      println("\nMenu:")
      println("1.View Products")
      println("2. Addto Cart")
      println("3.ViewCart")
      println("4.Exit")
```

```
    print("Enteryourchoice:")
    when(scanner.nextInt()){
        1->{

displayProducts(products)
        }
        2->{
            println("\nEntertheproductID toaddtocart:")
            valproductId=scanner.nextInt()
            valselectedProduct=products.find{ it.id==productId}
            if(selectedProduct!=null){
                addToCart(cart,selectedProduct)
            } else{
                println("InvalidproductID.Pleasetryagain.")
            }
        }
        3->{
            viewCart(cart)
        }
        4->{
            println("ThankyouforusingtheSimpleShoppingApp.Exiting...")
            return
        }
        else ->{
           println("Invalidchoice.Pleaseenteravalid option.")
        }
    }
  }
```

Sample output



RESULT:

Thus the shopping application was created and the products are viewed successfully

ExNo:8	web server supporting push notifications.
Date:	

Aim:

To design a web server supporting push notifications.

Procedure:

1) Create a new Kotlin project: Use your preferred IDE or build tool to create a new Kotlin project.
2) Add Ktor Dependencies: Open yourbuild.gradle.kts(orbuild.gradle) fileand include the following dependencies for Ktor:

```
plugins {
kotlin("jvm")vers
ion"1.5.31"
id("io.ktor")versi
on"1.6.10"
}
repositories{
mavenCentral()
}
dependencies{
implementation("io.ktor:ktor-server-netty:1.6.10")
implementation("io.ktor:ktor-websockets:1.6.10")
}
```

1) WriteKtorApplicationCode:Createanew Kotlinfile (e.g.,PushNotificationServer.kt) Andwrite the Ktorapplicationcode:
2) ExecutethemainfunctioninyourPushNotificationServer.ktfile.Thiswillstart theKtorserveronhttp://localhost:8080.CreateHTMLfile(e.g.,index.html)wit hasimpleWebSocket client:Opentheindex.htmlfileinawebbrowser. Youcanopenmultipleinstancesindifferent tabsor
3) browsers tosimulate multiple clients.

4) Interact with the WebSocket:Typeamessageinone client andclick "SendMessage." You should see the message pushed to all other connectedclientsinreal-time.

Program:

Kotlin:

```
import io.ktor.application.*
import
io.ktor.features.ContentNegotiationimportio.ktor.
features.StatusPages
importio.ktor.http.HttpStatusCode
importio.ktor.jackson.jacksonimporti
o.ktor.routing.routing
importio.ktor.server.engine.embeddedServer
importio.ktor.server.netty.Nettyimportio.kto
r.websocket.WebSocketsimport
io.ktor.websocket.webSocket
import kotlinx.coroutines.channels.ClosedReceiveChannelException
importkotlinx.coroutines.channels.consumeEachimportj
ava.time.Duration
funApplication.module(){
   install(ContentNegotiation){
      jackson{ }
   }
   install(StatusPages){
      exception<Throwable>{cause ->
```

```
        call.respondText(cause.localizedMessage,status=HttpStatusCode.InternalServerError)
      }
   }

   install(WebSockets){
          pingPeriod=Duration.ofSeconds(60)



timeout =Duration.ofSeconds(15)
   maxFrameSize=Long.MAX_VALUE masking=
   false
}
routing{
   valconnections=mutableListOf<DefaultWebSocketServerSession>()
   webSocket("/push"){connections.
      add(this) try {
         incoming.consumeEach{ frame->
                  if(frameisFrame.Text) {
            valmessage=frame.readText()
            connections.forEach{
                     if(it != this){
                  it.send(Frame.Text(message))
               }
            }
          }
         }
      }catch(e:ClosedReceiveChannelException){
         //Channelwasclosednormally
```

```
    } finally {
        connections.remove(this)
    }
  }
}
        }
        funmain(){
            embeddedServer(Netty,port =8080,module =Application::module).start(wait=true)
        }
```

HTMLfile:

```
<!DOCTYPEhtml>
<htmllang="en">
<head>
    <metacharset="UTF-8">
    <metaname="viewport"content="width=device-width,initial-scale=1.0">
    <title>WebSocketPushNotification</title>
</head>
<body>
    <inputtype="text"id="messageInput"placeholder="Typeamessage">
    <buttononclick="sendMessage()">SendMessage</button>
    <ulid="messages"></ul>
    <script>
        constsocket=newWebSocket("ws://localhost:8080/push");
        socket.onmessage=(event)=>{
            constmessages=document.getElementById("messages");
```

```
                const li=document.createElement("li");
                li.appendChild(document.createTextNode(event.data));
                messages.appendChild(li);
 };
        functionsendMessage(){
                constinput=document.getElementById("messageInput");
                const message=input.value;
                socket.send(message);
                input.value ="";
            }
          </script>
        </body>
</html>
```

Sample output

Result

Thus the web server supporting push notifications was implemented successfully.

ExNo:9	Develop an application by integrating Google maps
Date:	

Aim:

To develop an application by integrating Google maps

Procedure:

To integrate Google Maps into an Android application using Kotlin,you can follow these steps:

GetAPI Key:

Visit the Google Cloud Console.

Create a new project or select an existingone.

Enable the"MapsSDK for Android" for yourproject.

Create an API key.

Add Google Maps Dependency:

Open your app-level build.gradle file and add the Google Maps dependency:

gradle

```
implementation 'com.google.android.gms:play-services-maps:17.0.1'
```

Add Permissions:

Make sure you have the necessary permissions in your Android Manifest.xmlfile:

xml

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

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

<uses-permissionandroid:name="android.permission.ACCESS_COARSE_LOCATION"/>
```

Add APIKey:

Add yourGoogle Maps APIkey to the Android Manifest.xml file:

Xml

```
<application>
   <!--Otherapplicationelements-->
```

```
<meta-data
        android:name="com.google.android.geo.API_KEY"
        android:value="YOUR_API_KEY"/>
</application>
```

CreateaMapFragment:

Create alayoutfile(fragment_map.xml) forthemapfragment:

Xml

```
<!--fragment_map.xml-->
<fragment android:id="@+id/mapFragment"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

InitializeGoogleMap:

Inyouractivityorfragment, initializetheGoogleMapobject:

kotlin

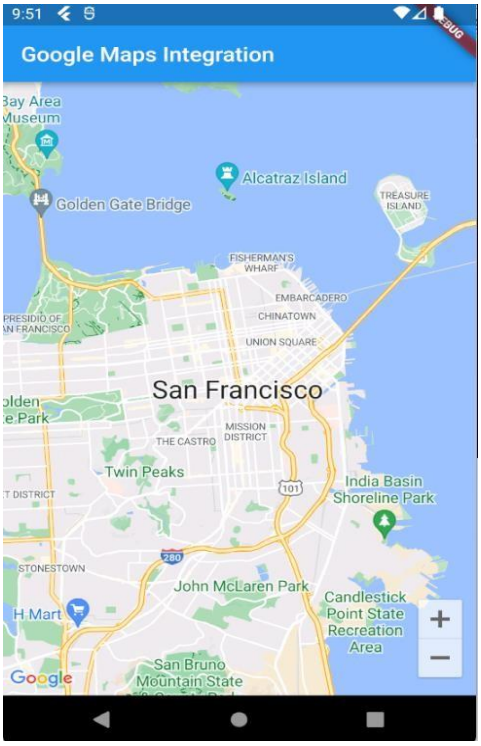
```
importcom.google.android.gms.maps.GoogleMap
importcom.google.android.gms.maps.OnMapReadyCallbacki
mportcom.google.android.gms.maps.SupportMapFragment
classMapFragment:Fragment(R.layout.fragment_map),OnMapReadyCallback{
    privatelateinitvargoogleMap:GoogleMap
    override funonViewCreated(view:View,savedInstanceState:Bundle?){
        super.onViewCreated(view, savedInstanceState)
        valmapFragment=childFragmentManager.findFragmentById(R.id.mapFragment)as
SupportMapFragment
        mapFragment.getMapAsync(this)
    }
```

```
    override funonMapReady(map:GoogleMap){
        googleMap=map
        //Addyourmapconfigurationshere
        //Forexample:setMapType,addMarkers,etc.
    }
}
```

RuntheApp:

Run your application on an emulator or a physical device to see the Google Map in action.

Sample output



Result

Thus the Google map application was designed and implemented successfully.

ExNo:10	Mini Project Using Flutter/Kotlin
Date:	

AIM

To Write a mobile applicationthat createsalarmclock.

PROCEDURE

Createa newAndroid Application

1. InEclipsegoto**File->New**->**Project**
2. Select an**AndroidProject** fromtheAndroidFolderandpress**Next.**
3. Fillinthe detailsofyourAndroidapplication.
   a. **ProjectName**:The project nameand folderthatEclipse willstore theprojectfiles
   b. **BuildTarget:**The versionoftheAndroidSDKthatwillbe usedwhen youbuild yourprogram.Selecta platform thatisequaltoorlowerthanthe targetchosen fortheAVD.
   c. **ApplicationName:**Thisis the name oftheapplication.
   d. **PackageName**:The namespace that allofthe source code willresideunder.
   e. **CreateActivity:**The name forthatclass stubthat isgenerated by theplugin.
4. Thevaluesthat are usedinthis exampleare:
   a. **ProjectName**:Alarm
   b. **BuildTarget:** 2.3.3
   c. **ApplicationName:**Alarm
   d. **PackageName**:com.Alarm.example
   e. **Create Activity:** Alarm

5. Click on

Finish.Coding

1. Open**AndroidMainfest.xml** whichislocatedin**res**->**values**->**AndroidMainfest.xml.** Thisfilewillholdallofthe text that ourlayout willuse.
2. Click onthe**AndroidMainfest**.xmlat the bottomtobringupthe raw xmlfile.

Editingthethejavacode

-

1. OpenSampleApp.java from theleft handside.
2. Savethe files.

RunningtheApplication

1. Click onthe greencirclewiththewhite arrow.
2. Choose theAVD that we created inapreviousstep.

PROGRAMS

FileName:MainActivity.java

```
Package
com.lab.alarmclock;
importjava.util.Calendar;
import
android.app.Activity;
import
android.app.AlarmManager;
import
android.app.PendingIntent;
import
android.content.Context;
importandroid.content.Intent;
importandroid.os.Bundle;
importandroid.view.View;
import
android.view.View.OnClickListener;
importandroid.widget.Button;
import android.widget.TimePicker;

 publicclassAlarmActivityextendsActivity
                                {

        privateTimePickertimep
        icker;privateContextcon
        text;privateButtonbtnS
        etAlarm;

        @Override
        protectedvoidonCreate(BundlesavedInstanceState){
                //TODOAuto-
                generatedmethodstubsuper.onCreate(s
                avedInstanceState);
                setContentView(R.layout.activity_main);

               context=this;

                timepicker=(TimePicker)findViewById(R.id.timepicker);
```

```
btnSetAlarm=(Button)findViewById(R.id.btnSetAlarm);
                                @Override
                                publicvoidonClick(Viewv){
                                        //TODOAuto-generatedmethodstub


                                        Calendarcalendar=Calendar.getInstance();
                                        calendar.set(Calendar.HOUR_OF_DAY,
                                        timepicker.getCurrentHour());calendar.set(Calendar.M
                                        INUTE, timepicker.getCurrentMinute());

                                        IntentmyIntent=new Intent(context,
                                        AlarmReceiver.class);PendingIntentpendingI
                                        ntent= PendingIntent.getBroadcast(
                                                        context,0,myIntent, 0);

                                        AlarmManageralarmMa
        nager=
        (AlarmManager)getSystemService(ALARM_SERVICE);
                                        alarmManager.set(AlarmManager.RTC,calendar.getTimeIn
                                        Millis(),
                                                        pendingI
                                                         ntent);


                                }
                        });

                };}


 }
```

FileName: AlaramReciever.java

```
packagecom.lab.alarmclock;

import
android.content.C
ontext; import
android.content.I
ntent; import
```

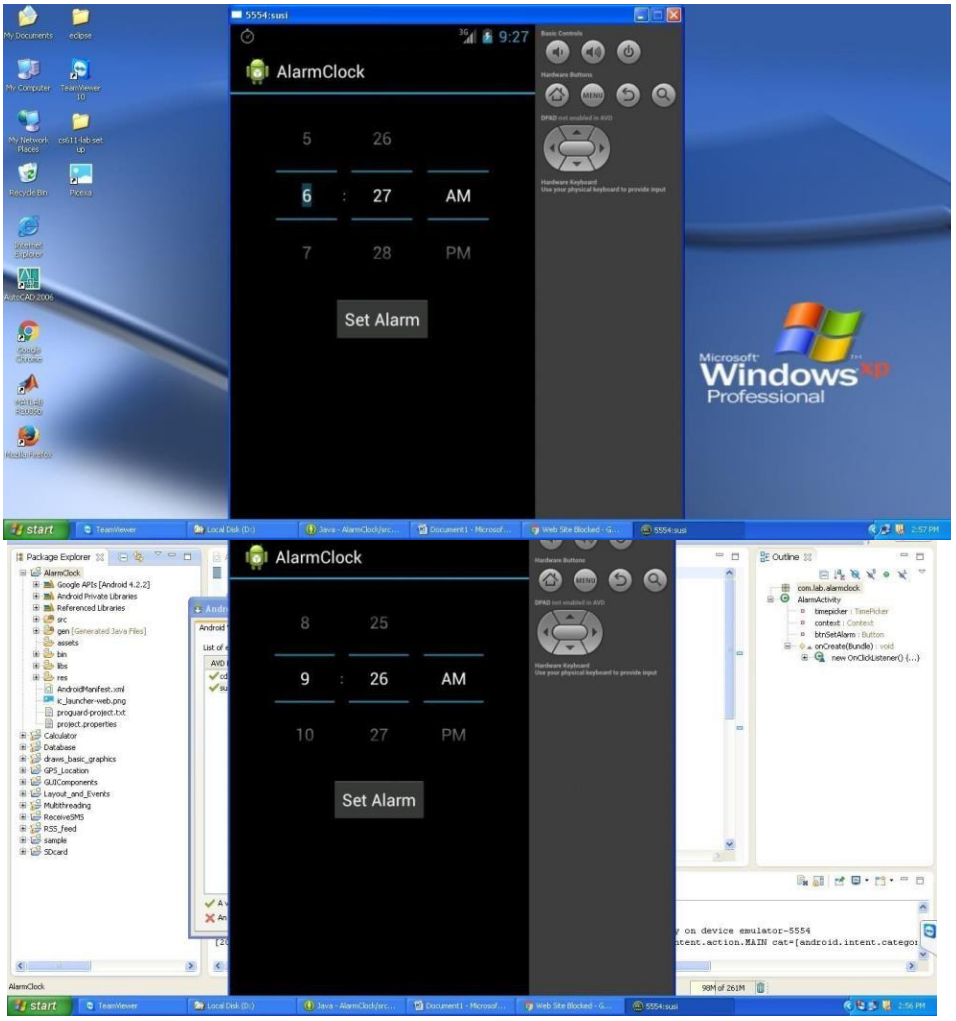
```
    android.media.Ri
    ngtone; import
    android.media.RingtoneManager;

    import android.net.Uri;
mportandroid.support.v4.content.WakefulBroadcastRe
ceiver; import android.util.Log;
import android.widget.Toast;

publicclassAlarmReceiverextends

        WakefulBroadcastReceiver{ @Override
          publicvoidonReceive(Contextcontext,Intentintent){
                //TODOAuto-generatedmethodstub

                Log.e("alarmreceiver","alarmreceiver");
                Toast.makeText(context,"alarmreceiver",Toast.LENGTH_LONG).show();

                UrialarmUri=RingtoneManager
                                .getDefaultUri(RingtoneManager.TYPE_ALARM);

                Ringtoneringtone=RingtoneManager.getRingtone(context,

                alarmUri);ringtone.play();

        }
}
```

FileName: Androidmainfest.xml

```
<?xmlversion="1.0"encoding="utf-8"?>

<manifest
   xmlns:android="http://schemas.android.com/apk/res/androi
   d"package="com.lab.alarmclock"

   android:versionCode="1"
   android:versionName="1.0"
   >

   <uses-permissionandroid:name="android.permission.WAKE_LOCK"/>
```

```
    <uses-
      sdkandroid:minSdkVersion="8"
      android:targetSdkVersion="21"
      /><application android:allowBackup="true "
      android:icon="@drawable/ic_launche                                    r"
      android:label="@string/app_name"

      android:theme="@style/AppTheme"
      >

      <activity
        android:name=".AlarmActivity"
        android:label="@string/app_name
        "
        >

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

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

        </intent-filter>
      </activity>
      <receiverandroid:name=".AlarmReceiver" />

    </application>

  </manifest>
```

Sample output



Result

Thus the mobile application using alaram clock was designed and implemented successfully.

CONTENT BEYONDSYLLABUS

Ex. No. 01	**Develop an application that uses Layout Managers and Event Listeners**
Date:	

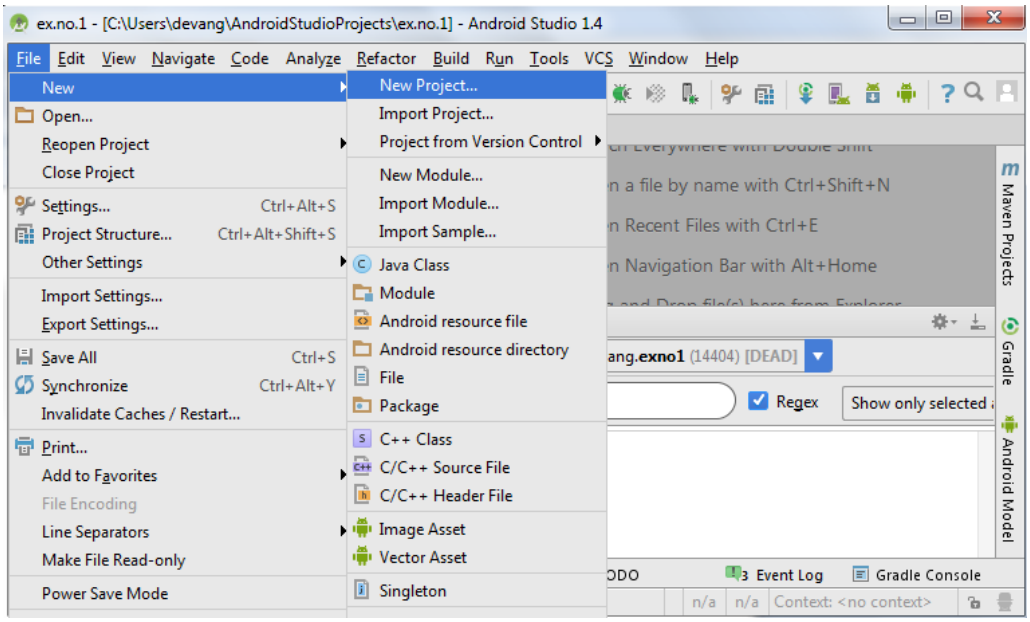
Aim:

To develop a Simple Android Application that uses Layout Managers and Event Listeners.

Procedure :

Creating a New project:

- Open Android Studio and then click on **File->New->Newproject**.



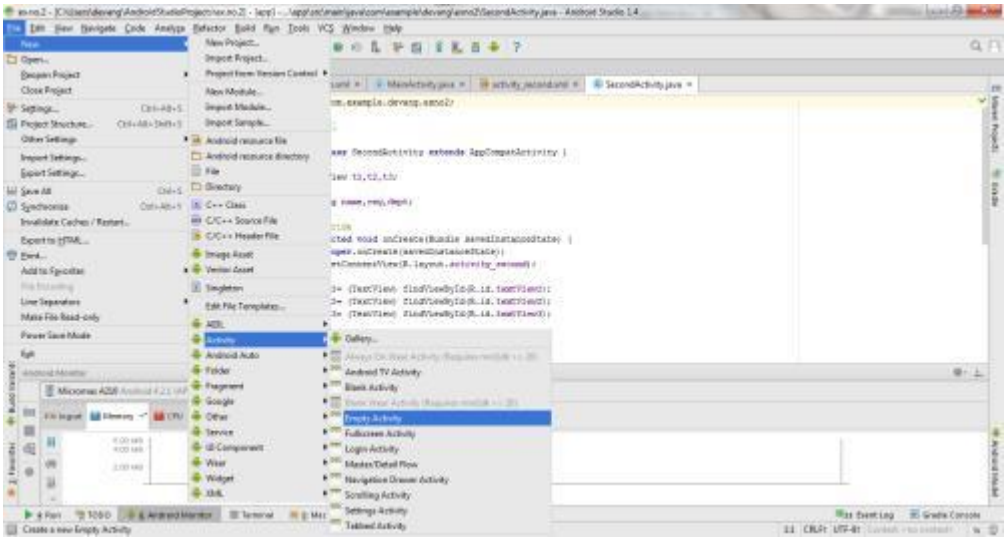
- Then type the Application name as**"exno2"**and click Next.
- Then **select the Minimum SDK** as shown below and click Next.
- Then **select the Empty Activity** and click Next.
- Finally click **Finish**.
- It will take some time to build andloadthe project.
- After completion it will Look as given below.

Creating Second Activity for the Android Application

- Then type the Application name as**"exno2"**and click Next.
- Then **select the Minimum SDK** as shown below and click Next.
- Then **select the Empty Activity** and click Next.
- Finally click **Finish**.
- It will take some time to build and load the project.
- After completion it will look as given below.

Creating Second Activity for the Android Application:

- Click on **File->New->Activity ->Empty Activity**.



- Type the Activity Name as**SecondActivity**andclick Finishbutton.
- ThusSecond Activity Forthe applicationiscreated.

DesigningLayoutforMain Activity:

- Clickon**app->res->layout ->activity_main.xml**.
- NowclickonText as shownbelow.
- Thendelete the code which isthere andtype thecode asgivenbelow.

CodeforActivity_main.xml:

```
<?xmlversion="1.0" encoding="utf-8"?>
<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">

<LinearLayout
     android:layout_width="match_parent"
     android:layout_height="100dp">
<TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        android:text="Details Form"
        android:textSize="25sp"
        android:gravity="center"/>
</LinearLayout>
```

```
<GridLayout android:id="@+id/gridLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="100dp"
    android:layout_marginBottom="200dp"
    android:columnCount="2"
    android:rowCount="3">
<TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_row="0"
      android:layout_column="0"
      android:text="Name"
      android:textSize="20sp"
      android:gravity="center"/>

<EditText android:id="@+id/editText"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_row="0"
      android:layout_column="1"
      android:ems="10"/>

<TextView android:id="@+id/textView2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_row="1"
      android:layout_column="0"
      android:text="Reg.No"
      android:textSize="20sp"
      android:gravity="center"/>
```

```
<EditText android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="1"
        android:layout_column="1"
        android:inputType="number"
        android:ems="10"/>

<TextView android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="2"
        android:layout_column="0"
        android:text="Dept"
        android:textSize="20sp"
        android:gravity="center"/>
<Spinner
        android:id="@+id/spinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="2"
        android:layout_column="1"
        android:spinnerMode="dropdown"/>
</GridLayout>
<Button android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentBottom="true"
      android:layout_centerInParent="true"
      android:layout_marginBottom="150dp"
      android:text="Submit"/>

</RelativeLayout>
```

DesigningLayoutforSecondActivity:

- Clickon**app->res->layout ->activity_second.xml**.
- NowclickonText as shownbelow.
- Thendelete the code which isthere andtype thecode asgivenbelow.

CodeforActivity_second.xml:

```
<?xmlversion="1.0"encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context="com.example.devang.exno2.SecondActivity" android:
  orientation="vertical"
  android:gravity="center">

  <TextView android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:text="NewText"
    android:textSize="30sp"/>

  <TextView android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:text="NewText"
    android:textSize="30sp"/>

  <TextView android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:text="NewText"
    android:textSize="30sp"/>
</LinearLayout>
```

- NowclickonDesign andyouractivity willlook asgivenbelow.
- Sonowthedesigningpartof Second Activity isalsocompleted.

Java Codingforthe AndroidApplication:

- JavaCoidngforMainActivity:
- Clickon**app->java->com.example.exno2 ->MainActivity**.
- Thendelete the code which isthere andtype thecode asgivenbelow.

CodeforMainActivity.java:

```
package com.example.exno2;

importandroid.content.Intent;
//importandroid.support.v7.app.AppCompatActivity;
import android.os.Bundle;
importandroid.view.View;
importandroid.widget.ArrayAdapter;
importandroid.widget.Button;
importandroid.widget.EditText;
importandroid.widget.Spinner;
importandroidx.appcompat.app.AppCompatActivity;

public classMainActivity extendsAppCompatActivity {

  //Definingthe Views
  EditTexte1,e2;
  Buttonbt;
  Spinners;

  //Dataforpopulating inSpinner
  String[]dept_array={"CSE","ECE","IT","Mech","Civil"};

  Stringname,reg,dept;

  @Override
  protectedvoidonCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //Referringthe Views
    e1=(EditText) findViewById(R.id.editText);
```

```
    e2=(EditText) findViewById(R.id.editText2);

    bt=(Button) findViewById(R.id.button);

    s=(Spinner) findViewById(R.id.spinner);

    //CreatingAdapterforSpinnerforadaptingthedatafromarraytoSpinner
    ArrayAdapteradapter= new
    ArrayAdapter(MainActivity.this,android.R.layout.simple_spinner_item,dept_array);
    s.setAdapter(adapter);

    //CreatingListenerfor Button
    bt.setOnClickListener(newView.OnClickListener() {
      @Override
      public voidonClick(Viewv) {

        //GettingtheValuesfromViews(Edittext&Spinner)
        name=e1.getText().toString();
        reg=e2.getText().toString();
        dept=s.getSelectedItem().toString();

        //IntentForNavigatingtoSecond Activity
        Intenti=newIntent(MainActivity.this,SecondActivity.class);

        //ForPassingtheValuestoSecond Activity
        i.putExtra("name_key",name);
        i.putExtra("reg_key",reg);
        i.putExtra("dept_key",dept);

        startActivity(i);
      }
    });
  }
}
```

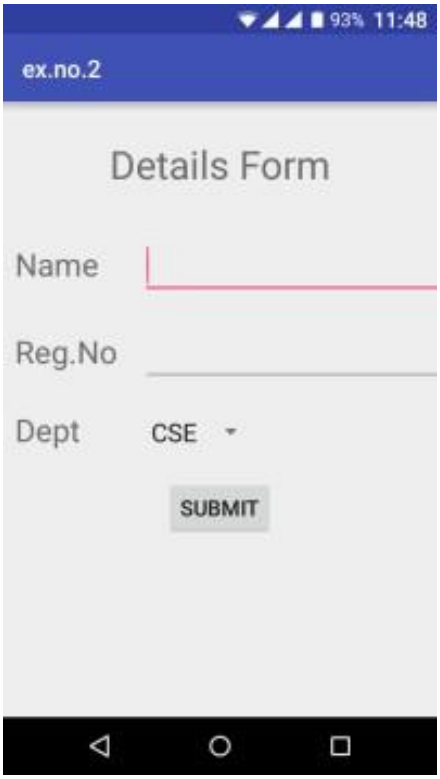
Java CodingforSecondActivity:

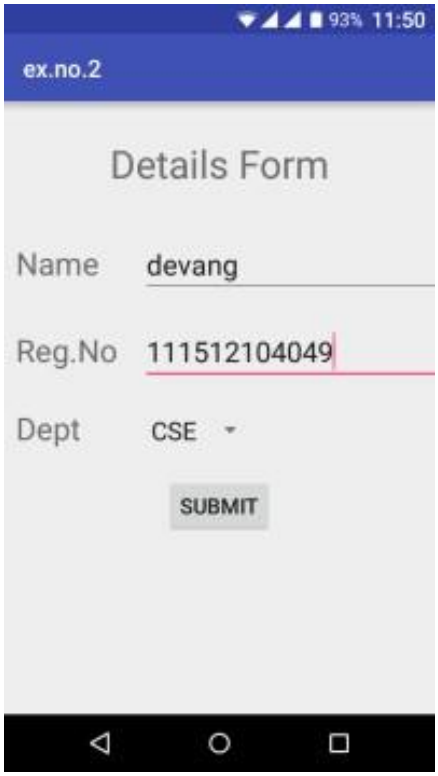
- Clickon**app->java->com.example.exno2 ->SecondActivity**.
- Thendeletethe code whichisthereandtypethecodeasgivenbelow.

CodeforSecondActivity.java:

```
package com.example.exno2;

importandroid.content.Intent;
//import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
importandroid.widget.TextView;
importandroidx.appcompat.app.AppCompatActivity;

public classSecondActivityextendsAppCompatActivity{
  TextViewt1,t2,t3;
  Stringname,reg,dept;
  @Override
  protectedvoidonCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);
    t1=(TextView)   findViewById(R.id.textView1);
    t2=(TextView)   findViewById(R.id.textView2);
    t3=(TextView) findViewById(R.id.textView3);
    //GettingtheIntent
    Intenti=getIntent();
    //GettingtheValuesfromFirstActivityusingtheIntentreceivednam
    e=i.getStringExtra("name_key");
    reg=i.getStringExtra("reg_key");
    dept=i.getStringExtra("dept_key");
    //SettingtheValuestoIntent
    t1.setText(name);
    t2.setText(reg);
    t3.setText(dept);
  }
}
```

- SonowtheCodingpartofSecond Activity isalsocompleted.
- Nowruntheapplicationtoseetheoutput.





Result:

Thus the **Layout Managers and Event Listeners application was implemented and verified successfully.**

Ex No. 02	**Write an application that draws Basic Graphical Primitives on the screen**
Date:	

Aim:

To develop a Simple Android Application that draws basic Graphical Primitives on the screen.

Procedure:

Creating a Newproject:

- Open Android Studio and then click on **File->New->Newproject**.
- Then type the Application name as**"exno3"**and click Next.
- Then **select the Minimum SDK** as shown below and click Next.
- Then **select the Empty Act**ivity and click Next.
- Finally click **Finish**.
- It will take some time to build and load the project.
- After completion it will look as given below.

Designing layout for the Android Application:

- Click on **app->res->layout ->activity_main.xml**.
- Now click on Text as shown below.
- Then delete the code which is there and type the code as given below.

CodeforActivity_main.xml:

```
<?xmlversion="1.0" encoding="utf-8"?>
<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent">

<ImageViewandroid:layout_width="match_
    parent"
    android:layout_height="match_parent"
    android:id="@+id/imageView" />
</RelativeLayout>
```

- NowclickonDesignandyourapplicationwilllookasgivenbelow.
- Sonowthedesigningpartiscompleted.

Java Coding for the Android Application:

- Click on **app->java->com.example.exno3 ->MainActivity**.
- Then delete the code which is there and type the code as given below.

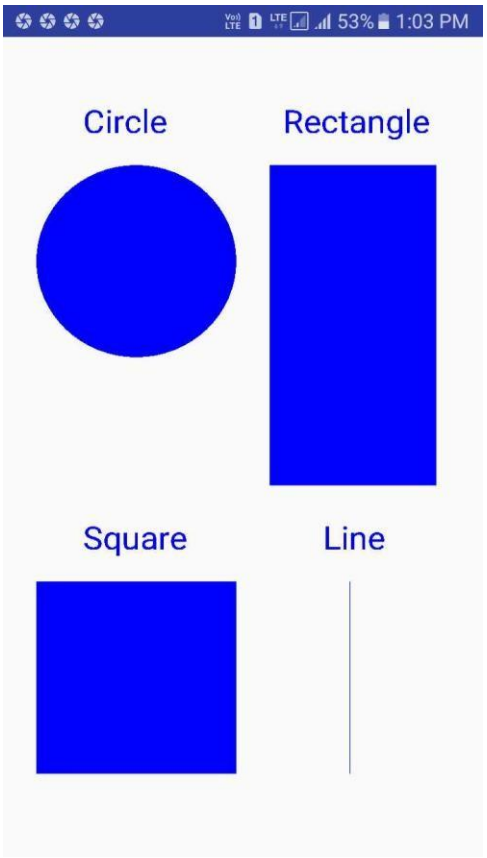
CodeforMainActivity.java:

```
package com.example.exno3;

importandroid.app.Activity;
importandroid.graphics.Bitmap;
importandroid.graphics.Canvas;
importandroid.graphics.Color;
importandroid.graphics.Paint;
importandroid.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
importandroid.widget.ImageView;

public classMainActivityextendsActivity
{
   @Override
   public voidonCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      //Creating a Bitmap
      Bitmapbg=Bitmap.createBitmap(720,1280,Bitmap.Config.ARGB_8888);

      //Setting   the Bitmap as background for the Image ViewI
      mage View i=(ImageView)  find View ById(R.id.imageView);
      i.setBackgroundDrawable(newBitmapDrawable(bg));

      //Creating the Canvas Object
      Canvascanvas=newCanvas(bg);

      //Creating the Paint Object and set its color &TextSize
      Paintpaint=newPaint();
      paint.setColor(Color.BLUE);
      paint.setTextSize(50);

      //TodrawaRectanglecanvas.drawText("Rectan
      gle",420,150,paint);
      canvas.drawRect(400,200,650,700,paint);

      //Todrawa Circle
      canvas.drawText("Circle",120,150,paint);
```

```
    canvas.drawCircle(200,350,150,paint);

    //TodrawaSquarecanvas.drawTe
    xt("Square",120,800,paint);
    canvas.drawRect(50,850,350,1150,paint);

    //TodrawaLine
    canvas.drawText("Line",480,800,paint);
    canvas.drawLine(520,850,520,1150,paint);
  }
}
```

- SonowtheCodingpartisalsocompleted.
- Nowruntheapplicationtoseetheoutput.

Sample output



Result:

Thus the graphical primitives was designed and implemented successfully.

Ex. No.03 **Develop an application that makes use of database**
Date:

Aim:

To develop a Simple Android Application that makes use of Database.

Procedure:

CreatingaNewproject:

- OpenAndroidStudioandthenclickon**File->New->Newproject**.
- ThentypetheApplication name as**"exno4"**andclickNext.
- Then **selecttheMinimum S**DKasshownbelowandclickNext.
- Then **selecttheEmpty Activity**andclickNext.
- Finally click **Finish**.
- Itwilltake some time tobuildandloadtheproject.
- Aftercompletionitwilllookasgivenbelow.

DesigninglayoutfortheAndroidApplication:

- Clickon**app->res->layout ->activity_main.xml**.
- NowclickonText asshownbelow.
- Thendeletethe code whichisthereandtypethecodeasgivenbelow.

CodeforActivity_main.xml:

```
<?xmlversion="1.0" encoding="utf-8"?>
<AbsoluteLayoutxmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
<TextViewandroid:layout_width="wrap_co
    ntent"
    android:layout_height="wrap_content"
    android:layout_x="50dp"
    android:layout_y="20dp"
    android:text="StudentDetails"
    android:textSize="30sp" />

<TextViewandroid:layout_width="wrap_co
    ntent"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="110dp"
    android:text="EnterRollno:"
```

```
<EditTextandroid:id="@+id/Rollno"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="100dp"
    android:inputType="number"
    android:textSize="20sp" />

<TextViewandroid:layout_width="wrap_co
    ntent"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="160dp"
    android:text="EnterName:"
    android:textSize="20sp" />

<EditText android:id="@+id/Name"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="150dp"
    android:inputType="text"
    android:textSize="20sp" />

<TextViewandroid:layout_width="wrap_co
    ntent"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="210dp"
    android:text="EnterMarks:"
    android:textSize="20sp" />

<EditText android:id="@+id/Marks"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
```

```
    android:layout_x="175dp"
    android:layout_y="200dp"
    android:inputType="number"
    android:textSize="20sp" />

<Button android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="300dp"
    android:text="Insert"
    android:textSize="30dp" />

<Button android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:text="Delete"
    android:textSize="30dp" />

<Button android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="400dp"
    android:text="Update"
    android:textSize="30dp" />

<Button android:id="@+id/View"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="400dp"
    android:text="View"
```

```
    android:textSize="30dp" />

<Button android:id="@+id/ViewAll"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_x="100dp"
    android:layout_y="500dp"
    android:text="ViewAll"
    android:textSize="30dp" />

</AbsoluteLayout>
```

- NowclickonDesignandyourapplicationwilllookasgivenbelow.
- Sonowthedesigningpartiscompleted.

Java CodingfortheAndroidApplication:

- Clickon**app->java->com.example.exno4 ->MainActivity**.
- Thendeletethe code whichisthereandtypethecodeasgivenbelow.

CodeforMainActivity.java:

```
packagecom.example.exno4;
 importandroid.app.Activity;
importandroid.app.AlertDialog.Builder;
importandroid.content.Context;
importandroid.database.Cursor;
importandroid.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
importandroid.view.View;
importandroid.view.View.OnClickListener;
importandroid.widget.Button;
importandroid.widget.EditText;

public classMainActivityextendsActivityimplementsOnClickListener
{
  EditTextRollno,Name,Marks;
  ButtonInsert,Delete,Update,View,ViewAll;
  SQLiteDatabasedb;
  /**Calledwhentheactivity isfirstcreated.*/
  @Override
  public voidonCreate(Bundle savedInstanceState)
  {
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);

    Rollno=(EditText)findViewById(R.id.Rollno);
    Name=(EditText)findViewById(R.id.Name);
    Marks=(EditText)findViewById(R.id.Marks);
    Insert=(Button)findViewById(R.id.Insert);
    Delete=(Button)findViewById(R.id.Delete);
    Update=(Button)findViewById(R.id.Update)
    ; View=(Button)findViewById(R.id.View);
    ViewAll=(Button)findViewById(R.id.ViewAll
    );

    Insert.setOnClickListener(this);
    Delete.setOnClickListener(this);
    Update.setOnClickListener(this)
    ; View.setOnClickListener(this);
    ViewAll.setOnClickListener(this
    );

    //Creatingdatabaseandtable
    db=openOrCreateDatabase("StudentDB",Context.MODE_PRIVATE,null);
    db.execSQL("CREATETABLEIFNOTEXISTSstudent(rollnoVARCHAR,nameVARCHAR,marks
VARCHAR);");
  }
  public voidonClick(Viewview)
  {
    //InsertingarecordtotheStudent table
    if(view==Insert)
    {
      //Checkingforempty fields
      if(Rollno.getText().toString().trim().length()==o
      ||
          Name.getText().toString().trim().length()==o||
          Marks.getText().toString().trim().length()==o)
      {
        showMessage("Error","Pleaseenterallvalues");
        return;
      }
      db.execSQL("INSERTINTOstudentVALUES('"+Rollno.getText()+"','"+Name.getText()+
          "','"+Marks.getText()+"');");
      showMessage("Success","Recordadded");
```

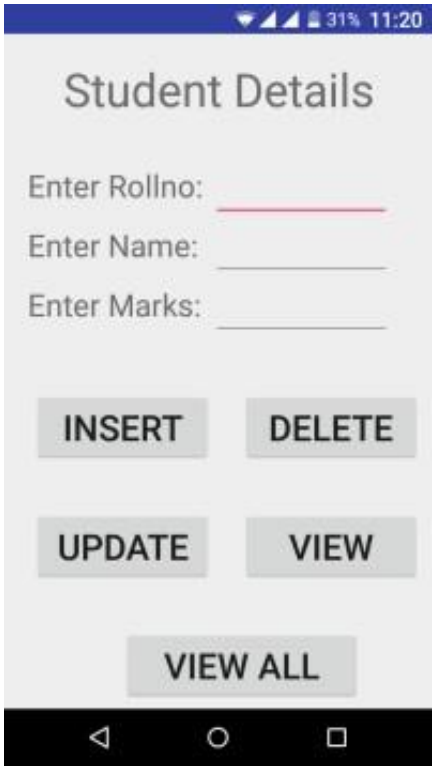
```
        clearText();
      }

      //Deletingarecordfrom the Student table
      if(view==Delete)
{
   //Checkingforemptyrollnumberif(Rollno.getTe
   xt().toString().trim().length()==o)
   {
      showMessage("Error","PleaseenterRollno");
      return;
   }
   Cursorc=db.rawQuery("SELECT* FROM studentWHERErollno='"+Rollno.getText()+"'",null);
   if(c.moveToFirst())
   {
      db.execSQL("DELETEFROM studentWHERErollno='"+Rollno.getText()+"'");
      showMessage("Success","RecordDeleted");
   }
   else
   {
      showMessage("Error","InvalidRollno");
   }
   clearText();
}
//UpdatingarecordintheStudent table
if(view==Update)
{
   //Checkingforemptyrollnumberif(Rollno.getTe
   xt().toString().trim().length()==o)
   {
      showMessage("Error","PleaseenterRollno");
      return;
   }
   Cursorc=db.rawQuery("SELECT* FROM studentWHERErollno='"+Rollno.getText()+"'",null);
   if(c.moveToFirst()) {
      db.execSQL("UPDATEstudentSETname='" +Name.getText() +"',marks='" +Marks.getText() +
            "' WHERErollno='"+Rollno.getText()+"'");
      showMessage("Success","RecordModified");
   }
   else {
      showMessage("Error","InvalidRollno");
```

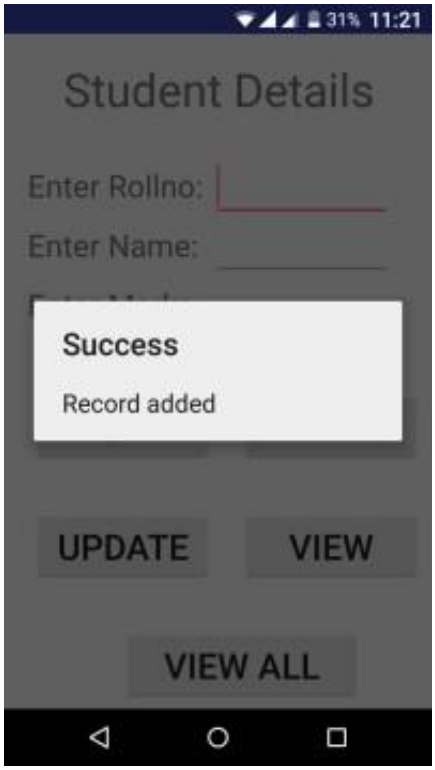
```
    }
    clearText();
}
//Display arecordfrom the Student table
if(view==View)
{
        //Checkingforemptyrollnumberif(Rollno.getText
        ().toString().trim().length()==0)
        {
            showMessage("Error","PleaseenterRollno");
            return;
        }
        Cursorc=db.rawQuery("SELECT* FROM studentWHERErollno='"+Rollno.getText()+"'",null);
        if(c.moveToFirst())
        {
            Name.setText(c.getString(1));
            Marks.setText(c.getString(2));
        }
        else
        {
            showMessage("Error","InvalidRollno");
            clearText();
        }
    }
    //Displayingalltherecordsif(
    view==ViewAll)
    {
        Cursorc=db.rawQuery("SELECT* FROM student",null);
        if(c.getCount()==0)
        {
            showMessage("Error","Norecordsfound");
            return;
        }
        StringBufferbuffer=newStringBuffer();
        while(c.moveToNext())
        {
            buffer.append("Rollno:"+c.getString(0)+"\n");
            buffer.append("Name: "+c.getString(1)+"\n");
            buffer.append("Marks:"+c.getString(2)+"\n\n");
        }
        showMessage("StudentDetails",buffer.toString());
```

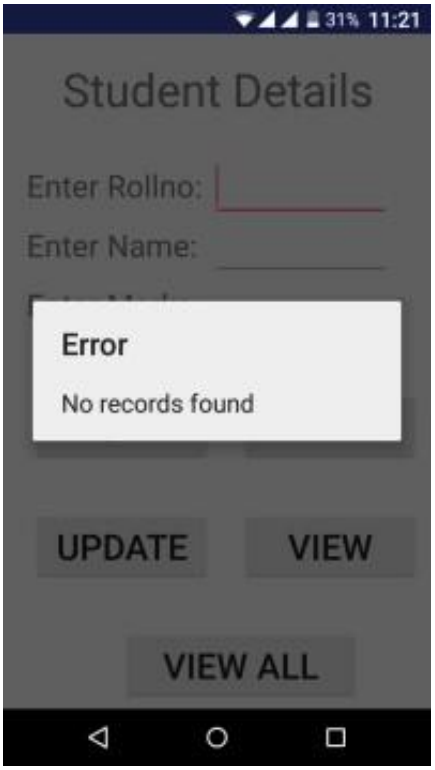
```
    }
}
public voidshowMessage(Stringtitle,Stringmessage)
{
    Builderbuilder=newBuilder(this);
    builder.setCancelable(true);
    builder.setTitle(title);



        builder.setMessage(message);
        builder.show();
    }
    public voidclearText()
    {
        Rollno.setText
        ("");
        Name.setText
        ("");
        Marks.setText
        ("");
        Rollno.request
        Focus();
    }
}
```

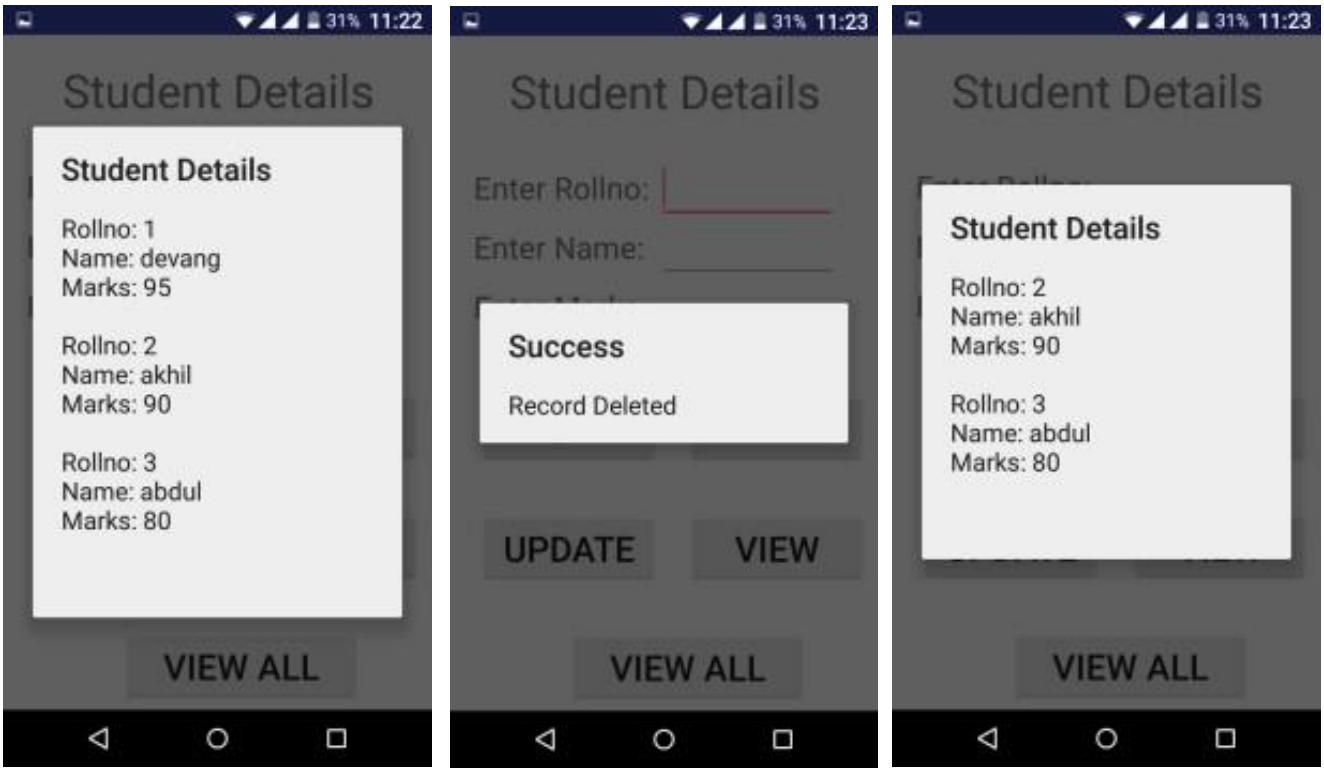
- SonowtheCodingpartisalsocompleted.
- Nowruntheapplicationtoseetheoutput.

Sample output









Result:

Thus the Database was created and the records are added , viewed and deleted successfully.

VIVAVOCE

EXERCISE 1

1. **What is Flutter?**

Flutter is an open-source UI software development kit created by Google. It is used to develop applications for mobile, web, and desktop from a single codebase.

2 **What are the advantages of using Flutter?**

- Single codebase for multiple platforms
- Fast development with hot reload
- Beautiful UI with customizable widgets
- Strong community support

3 **What is a widget in Flutter?**

In Flutter, everything is a widget, including layout elements, animations, and user interface components.

4**What is the difference between StatefulWidget and StatelessWidget?**

- StatelessWidget: Does not maintain any state and remains unchanged once built.
- StatefulWidget: Maintains state and updates UI dynamically.

5. How does Flutter handle state management?

Flutter supports various state management approaches, including:

- Provider
- Riverpod
- Bloc
- GetX

EXERCISE 2

1.What is pubspec.yaml file?

It is the project's configuration file that will use a lot during working with the Flutter project. It allows you how your application works. It also allows us to set the constraints for the app.

This file contains:

- Project general settings such as name, description, and version of the project.
- Project dependencies.
- Project assets (e.g., images, audio, etc.).

2. What are the advantages of Flutter?

The popular advantages of the Flutter framework are as follows:

- Cross-platform Development
- Faster Development
- Good Community
- Live and Hot Reloading
- Minimal code
- UI FocusedDocumentation

3.Why does the first Flutter app build take so long?

When you build the Flutter app the first time, it will take a longer time. It is because the Flutter built the device-specific APK or IPA file. Thus, the Gradle and Xcode are used to build the file, taking a long time.

4. Name some popular apps that use Flutter?

Some of the most popular app built on Flutter are as follows:

- Google Ads
- Reflectly
- Alibaba
- Birch Finance
- Coach Yourself
- Tencent
- Watermaniac

5. What is the latest release of Flutter SDK?
The latest release of the Flutter framework is Flutter- v1.20.4 on 15 September 2020.

EXERCISE 3

1. What is the primary purpose of a native calculator application?
To perform basic and advanced mathematical calculations

2. Which programming language is commonly used for developing native Android calculator applications?
Flutter,Android studio

3. What is Android Studio?
It's the official IDE for Android development, based on IntelliJ IDEA, used for building Android apps.

4. What is an Activity in Android?
An Activity is a single screen with a user interface. It acts as the entry point for interacting with the user.

5. What are Intents in Android?
Intents are messaging objects used to request actions from other app components (like starting another activity).

6. What is the difference between explicit and implicit intents?
Explicit: Directly specifies the target component.
Implicit: Declares a general action to be performed, resolved by the system.

EXERCISE 4

1. **What is the difference between Fragment and Activity?**

   Fragments are reusable components inside Activities; useful for modular UI and dynamic layouts.

2. **What is a View Model and why is it used?**

   Part of Android Jetpack; ViewModel stores and manages UI-related data in a lifecycle-conscious way.

3. **How does Android handle background tasks?**

   Through threads, AsyncTask (deprecated), WorkManager, or coroutines (in Kotlin).

4. **What is the use of Gradle in Android Studio?**

   Gradle is the build system used to automate tasks like compiling, testing, packaging, and deploying code.

5. **What is a Recycler View and how is it different from List View?**

   Recycler View is more advanced, efficient, and flexible than List View, supporting view recycling and layout managers.

EXERCISE 5

1. What is a Movie Rating Application?

A movie rating application is a platform where users can browse, search, rate, and review movies. It typically provides movie details, trailers, user ratings, critic scores, and personalized recommendations.

2.**What features should a basic movie rating application include?**

- User registration and login
- Movie browsing and search
- Rating and reviewing system
- Movie details (cast, release date, genre, etc.)
- Admin panel for managing content
- Recommendation system
- Sorting and filtering options

3.**How can you ensure the ratings are trustworthy?**

- Allow ratings only from verified users
- Use algorithms to detect spam or biased reviews
- Enable reporting of inappropriate content
- Aggregate multiple ratings (average score)
- Include both user and critic ratings

4. **How is the average movie rating calculated?**

The average rating is typically calculated by summing all user ratings and dividing by the number of ratings:
Average Rating = Sum of All Ratings / Total Number of Ratings

5. **What technologies are commonly used to build such an app?**

- **Frontend:** React, Angular, Flutter
- **Backend:** Node.js, Django, Spring Boot
- **Database:** MySQL, MongoDB, PostgreSQL
- **Authentication:** Firebase, OAuth **Hosting:** AWS, Heroku, Vercel

EXERCISE 6

1. What is a web server?
A web server is software (or hardware) that handles HTTP requests from clients (like browsers) and delivers web content (HTML, JSON, etc.) in response. Examples include Apache, Nginx, and Node.js.

2. What is the difference between frontend and backend in web development?
The **frontend** is the client-side (what users see — HTML, CSS, JS), and the **backend** is the server-side (handles logic, database interaction, APIs).

3. What is REST API?
REST (Representational State Transfer) is an architectural style for building web services. It uses HTTP methods like GET, POST, PUT, DELETE to access and modify resources.

4. What is middleware in web development?
Middleware is software that acts as a bridge between the server and application logic. In frameworks like Express.js, middleware functions handle requests, authentication, logging, etc.

5. **What are status codes in HTTP responses?**
Common codes include:

- 200 OK – Request successful
- 404 Not Found – Resource doesn't exist
- 500 Internal Server Error – Server-side issue
- 401 Unauthorized – Needs authentication
- 403 Forbidden – Access denied

EXERCISE 7

1. What are the key features of a successful shopping app?

- User registration & authentication
- Product catalog with filters and search
- Shopping cart &wishlist
- Secure payment gateway integration
- Order tracking
- Push notifications
- Reviews and ratings
- Admin panel for product & order management
- User analytics

2. Which tech stack is ideal for developing a shopping app?

- **Frontend (Mobile App):** Flutter, React Native, Swift (iOS), Kotlin (Android)
- **Backend:** Node.js, Django, Ruby on Rails, Laravel
- **Database:** PostgreSQL, MongoDB, MySQL
- **APIs:** RESTful APIs or GraphQL
- **Payment:** Stripe, Razorpay, PayPal
- **Hosting:** AWS, Firebase, Azure

3. How do you ensure the security of user data in a shopping app?

- Use HTTPS for secure communication
- Store sensitive data (like passwords) using encryption (e.g., bcrypt)
- Use token-based authentication (JWT)
- Regularly update dependencies and libraries
- Conduct security audits and penetration

4. What monetization strategies can be used in a shopping app?

Answer:

- Direct product sales
- Featured product listings for sellers
- In-app ads
- Subscription plans for premium features
- Affiliate marketing

5. How can performance be optimized in a shopping app?

- Lazy loading of images
- Efficient database queries and indexing
- Use of caching (e.g., Redis, CDN)
- Compress assets for faster loading
- Optimize app size and reduce memory usage

EXERCISE 8

1. What are push notifications in web development?
Push notifications are messages sent from a server to a client (web browser or mobile app) even when the client is not actively using the app or site. In web development, they are typically implemented using the Push API and Service Workers.

2. How do push notifications work in a web application?
Web push notifications involve these steps:

- User subscribes to notifications via the browser.
- The browser generates a subscription object with an endpoint and cryptographic keys.
- The backend server stores this subscription and sends push messages via a push service (like Firebase Cloud Messaging).
- A service worker receives the push message and displays it to the user.

3. What technologies are used to implement push notifications in web apps?

- Frontend: JavaScript, Push API, Notification API, Service Workers
- Backend: Node.js, Python, PHP, or any server-side language
- Push Services: Firebase Cloud Messaging (FCM), Web Push Protocol

4. What is a service worker?
A service worker is a background script that runs separately from the main web page, enabling features like push notifications, background sync, and offline functionality.

5. What is a push subscription object?
It's a JSON object returned when the user subscribes. It includes:

- endpoint URL
- expiration time
- public keys (p256dh and auth)

EXERCISE 9

1.How willyou integrate Google Maps into your mobile app?

To integrate Google Maps:

- For Android: Use the Google Maps SDK for Android and get an API key from Google Cloud Console.
- For iOS: Use the Google Maps SDK for iOS and follow similar steps to get your API key.
- You'll need to enable billing and the Maps SDK in your Google Cloud project.

2. What permissions is needed tosend request in your app?

You must request:

- ACCESS_FINE_LOCATION or ACCESS_COARSE_LOCATION (Android)
- Location When In Use or Always (iOS)
  These permissions allow the app to access the device's GPS location.

3. How can you show the user's current location on the map?

Once you have the correct permissions:

- Initialize the GoogleMap object.
- Use map.setMyLocationEnabled(true) in Android.
- Use GMSMapView with location services enabled for iOS.

4. How are the markersadded to a map?

Use the following snippet:

java

CopyEdit

```
LatLng location = new LatLng(latitude, longitude);

googleMap.addMarker(new MarkerOptions().position(location).title("Marker Title"));
```

5. Can you draw routes between two points?

Yes. Use the Google Directions API:

- Send an HTTP request with origin and destination.
- Parse the returned JSON polyline and draw it using Polyline Options.