Thadomal Shahani Engineering College

Bandra (W.), Mumbai- 400 050.

CERTIFICATE

Certify that ~~Mr.~~/Miss ISHIYA R. HARDASMANI

of COMPUTER Department, Semester VI with

Roll No. 2103058 has completed a course of the necessary

experiments in the subject MOBILE COMPUTING under my

supervision in the **Thadomal Shahani Engineering College**

Laboratory in the year 2023 - 2024

Teacher In-Charge

Head of the Department

Date 26/03/24

Principal

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

CONTENTS

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Experiment No. 1

(B)

Aim: write an android application (WAA) to draw basic graphical 2D primitives.

Theory: Android studio is official integrated development environment (IDE) for android application development. Android studio provides more features that enhance our productivity while building apps. It was announced on 16th may 2013 of Google I/O conference.

features of Android studio:

- → Flexible Gradle-based system
- → Fast & feature rich emulation for app testing.
- → Consolidated environment where we can develop for all android devices.
- → Supports C++ & NDK.
- → Supports Google cloud platform
- → Extensive testing tools & frameworks

Android studio project contains one or more modules with resource file & source code file. These include different types of modules

- → Android app modules
- → Library modules
- → Google app engine modules

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

- Folder structure
  - app
    - manifests (Manifest folder)
    - Java (Java folder)
      - com.tutlane.helloworld
      - com.tutlane.helloworld (android test)
      - com.tutlane.helloworld. (test
    - res (Resource folder)
      - drawable (Drawable folder)
      - layout (Layout folder)
      - mipmap (Mipmap folder)
      - values. (values folder)
    - Gradle scripts (Gradle folder).
      - build.gradle
      - build.gradle.
      - settings.gradle.

- functions.

1. Bitmap: It is a representation of graphic image as a pixel array. It is often used to load, manipulate & display images in Android apps Bitmap can be created from various sources including resources, files or dynamically generated. It takes width & height color format as parameter.

2. Canvas: The canvas class is used to draw graphics on android devices. It provides methods to draw various shapes, text & images on to bitmap or screen.

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

3- Image view: An image view is a UI widget in Android used to display images. It can be used to load & display bitmap, drawable or other image sources.

4. Paint: The 'paint' class is used to define how to draw graphical elements such as color, style, etc.

5. Color: The 'color' class in android provides method & work with colors.

Conclusion: In this experiment, we learnt about the tool android studio & how to draw 2D primitives in Android studio.

EXPERIMENT - 1

CODE:

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.drawable.BitmapDrawable;
import android.widget.ImageView;

import android.os.Bundle;


public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

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

        // Setting the Bitmap as background for the ImageView
        ImageView i = findViewById(R.id.imageview);
        i.setImageBitmap(bg);

        //Creating the Canvas Object
        Canvas canvas = new Canvas(bg);

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

        Paint paint1 = new Paint();
        paint1.setColor(Color.RED);
```

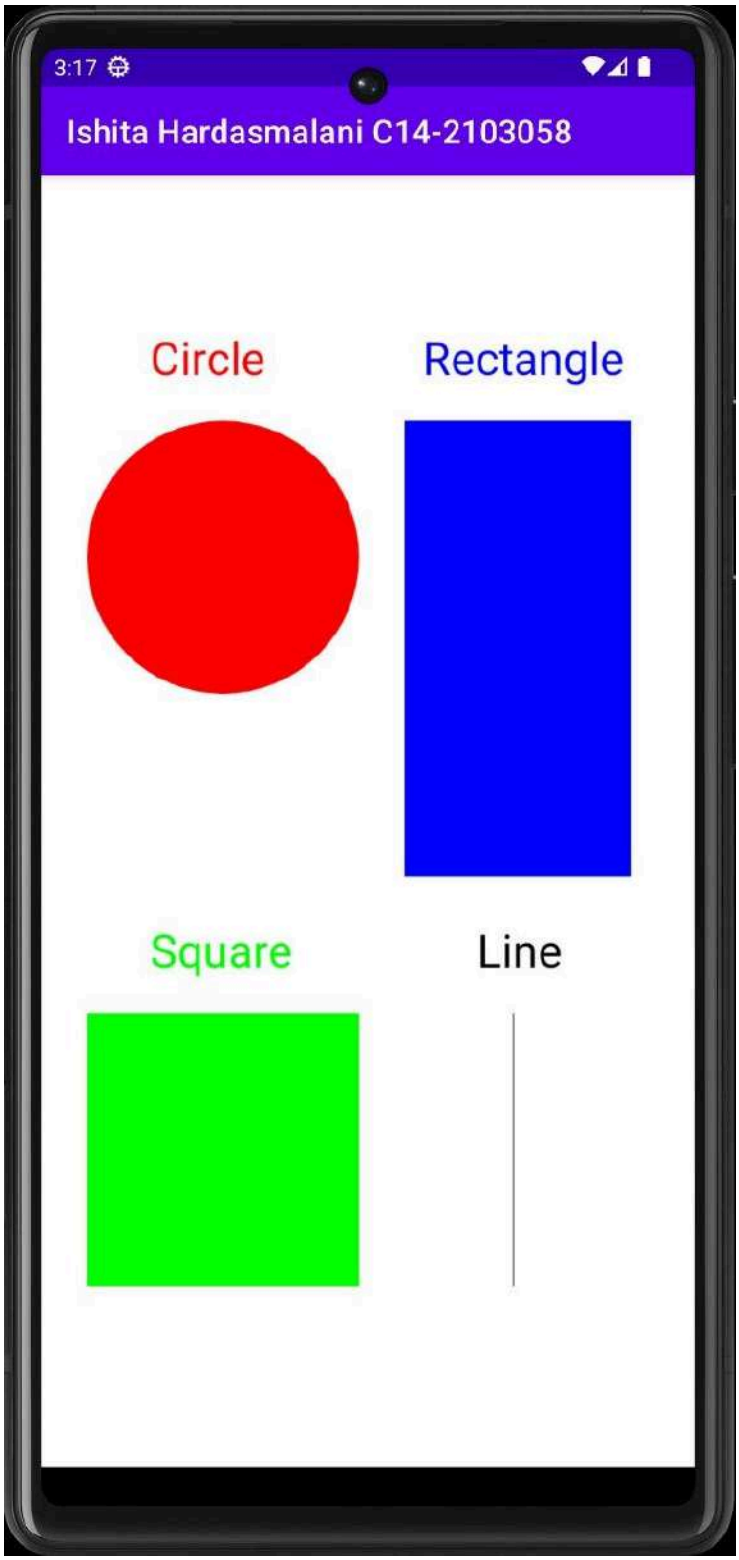
```
        paint1.setTextSize(50);

        Paint paint2 = new Paint();
        paint2.setColor(Color.GREEN);
        paint2.setTextSize(50);

        Paint paint3 = new Paint();
        paint3.setColor(Color.BLACK);
        paint3.setTextSize(50);

        Paint paint4 = new Paint();
        paint4.setColor(Color.BLACK);
        paint4.setTextSize(30);
        canvas.drawText("Yash Sabhandasani - 2103151",100, 60, paint4);

        //To draw a Rectangle
        canvas.drawText("Rectangle", 420, 150, paint);
        canvas.drawRect(400, 200, 650, 700, paint);

        //To draw a Circle
        canvas.drawText("Circle", 120, 150, paint1);
        canvas.drawCircle(200, 350, 150, paint1);

        //To draw a Square
        canvas.drawText("Square", 120, 800, paint2);
        canvas.drawRect(50, 850, 350, 1150, paint2);

        //To draw a Line
        canvas.drawText("Line", 480, 800, paint3);
        canvas.drawLine(520, 850, 520, 1150, paint3);
    }
}
```

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="MainActivity">

    <ImageView
      android:id="@+id/imageview"
      android:layout_width="match_parent"
      android:layout_height="match_parent" />

  </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:


3:17
Ishita Hardasmalani C14-2103058
Circle
Rectangle
Square
Line

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Experiment No. 2.

(13)

Aim: write an Android application to draw basic graphical 3D primitives

Theory:

This android program, written in Java, creates a simple 3D shape drawing application using the canvas & bitmap classes. Here is the detailed explanation

1. Package & Imports: This program is part of the 'com.example.a2dshape' package. Imports necessary classes for android app development including 'AppCompatActivity', 'Bitmap', 'Canvas', 'Color', 'Paint', 'Bundle', & 'ImageView'.
2. class & variables: The main activity class is 'MainActivity' extending 'AppCompatActivity'. Declares Bitmap 'bg' & ImageView 'img' as class variables
3. OnCreate Method: Initializes the activity sets the content view from 'activity_main.xml'. Creates a Bitmap 'bg' with dimensions 720x1280 pixels & sets it as the background for an ImageView. Initializes a canvas object associated with the Bitmap & fills it with a white color. Creates a paint object ('paint') for drawing shapes, sets color, stroke width & style.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

4. Drawing methods:
'draw 3D cube', 'draw 3D cuboid', 'draw 3D cylinder' & 'draw 3D cone' are ~~print~~ private methods for drawing 3D shapes on the canvas. Each method takes parameters specifying the position & dimensions of the shape, along with a paint object for styling. The cube, cuboid, cylinder & cone are drawn by combining, rectangles, ellipses & connecting lines, giving a simple 3D appearance.

5. Method usage: In 'onCreate' these methods are called with specific parameters to draw a 3D cube, cuboid, cylinder & cone on the canvas

Conclusion: This program essentially creates a basic 3D shape drawing app in Android, illustrating the use of canvas, Bitmap & paint for Graphical rendering.

EXPERIMENT - 02

CODE:

```
package com.example.myapplication;
import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {
    Bitmap bg;
    ImageView img;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // create bitmap
        bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.RGB_565);
        img = findViewById(R.id.imageView);
        img.setImageBitmap(bg);

        // canvas creation
        Canvas canvas = new Canvas(bg);
        canvas.drawColor(Color.WHITE);

        // paint object creation
        Paint paint1 = new Paint();
        paint1.setColor(Color.GREEN);
        paint1.setStrokeWidth(5); // Set the line thickness
        paint1.setStyle(Paint.Style.STROKE); // Set to stroke (no fill)

        Paint paint2 = new Paint();
        paint2.setColor(Color.BLACK);
        paint2.setTextSize(30);

        canvas.drawText("Pratham Rohra - 210315", 60, 80, paint2);
        // draw 3D cube
        draw3DCube(canvas, 450, 200, 650, 400, paint1);
```

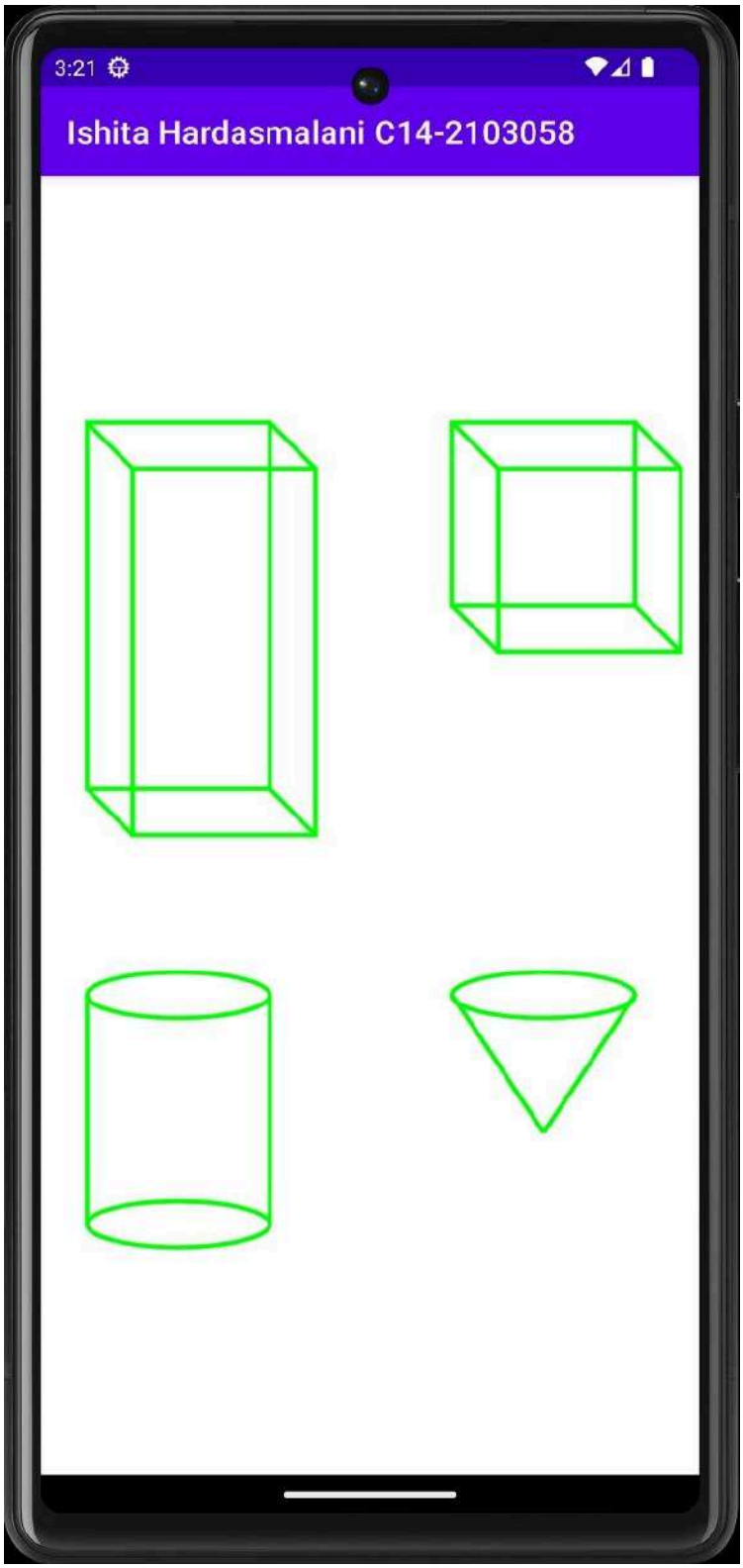
```
        draw3DCuboid(canvas, 50, 200, 250, 600, paint1);
        draw3DCylinder(canvas, 50, 800, 250, 1100, paint1);
        draw3Dcone(canvas, 450, 800, 650, 1000, paint1);
    }

    private void draw3DCube(Canvas canvas, float left, float top, float right, float bottom, Paint
paint) {
        // Draw front face border
        canvas.drawRect(left, top, right, bottom, paint);

        // Draw back face border
        canvas.drawRect(left + 50, top + 50, right + 50, bottom + 50, paint);

        // Draw connecting lines for perspective
        canvas.drawLine(left, top, left + 50, top + 50, paint);
        canvas.drawLine(right, top, right + 50, top + 50, paint);
        canvas.drawLine(left, bottom, left + 50, bottom + 50, paint);
        canvas.drawLine(right, bottom, right + 50, bottom + 50, paint);
    }

    private void draw3DCuboid(Canvas canvas, float left, float top, float right, float bottom, Paint
paint) {
        // Draw front face border
        canvas.drawRect(left, top, right, bottom, paint);

        // Draw back face border
        canvas.drawRect(left + 50, top + 50, right + 50, bottom + 50, paint);

        // Draw connecting lines for perspective
        canvas.drawLine(left, top, left + 50, top + 50, paint);
        canvas.drawLine(right, top, right + 50, top + 50, paint);
        canvas.drawLine(left, bottom, left + 50, bottom + 50, paint);
        canvas.drawLine(right, bottom, right + 50, bottom + 50, paint);
    }

    private void draw3DCylinder(Canvas canvas, float left, float top, float right, float bottom, Paint
paint) {
        // Draw top ellipse
        canvas.drawOval(left, top, right, top + 50, paint);

        // Draw bottom ellipse
        canvas.drawOval(left, bottom - 50, right, bottom, paint);

        // Draw connecting lines
```

```
        canvas.drawLine(left, top + 25, left, bottom - 25, paint);
        canvas.drawLine(right, top + 25, right, bottom - 25, paint);
    }

    private void draw3Dcone(Canvas canvas, float left, float top, float right, float bottom, Paint
paint) {
        // Draw top ellipse
        canvas.drawOval(left, top, right, top + 50, paint);

        // Draw connecting lines
        canvas.drawLine(left, top + 25, left+100, bottom - 25, paint);
        canvas.drawLine(right, top + 25, left+100, bottom - 25, paint);
    }
}
```

XML:

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

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView"/>
</LinearLayout>
```

OUTPUT:



THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

EXPERIMENT NO. 3. (B)

Aim: write an android application to design a form with GUI components.

Theory:

This code is for a simple android application developed using Java. It defines the main activity class which extends 'AppCompatActivity', a base class for activities that use the support library action bar features.

functionalities:

1. variable declarations: the class declares private variables for UI components. the 'EditText' fields for the username, surname, email & age, 'Button' for submitting the form.

2. onCreate Method: This is an overridden method from the 'AppCompatActivity' class which is called when the activity is first created. It performs essential startup tasks for the activity.
   → sets the content view to 'activity_main', which is a resources XML layout file defining the UI structure.
   → initializes the previous declared variables by finding the UI components in the layout using their IDs.

This is done with the 'findViewById' method, which locates views based on their identifiers in the xml file.

3] Button Click Listener:
An 'onClickListener' is set on the 'buttonSubmit' when the button is clicked, the listeners 'onClick' method is invoked. Inside this method, it performs the following actions:
→ Retrieves text values from the 'EditText' field name & email.
→ stores these values into a greeting message string.
→ Makes a toast of this string & displays to the user:
Toast.makeText().show() function.

EXPERIMENT 03

CODE:

```
package com.example.myapplication;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText editTextFirstName, editTextLastName, editTextEmail, editTextAge;
    private Button buttonSubmit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editTextFirstName = findViewById(R.id.editTextFirstName);
        editTextLastName = findViewById(R.id.editTextLastName);
        editTextEmail = findViewById(R.id.editTextEmail);
        editTextAge = findViewById(R.id.editTextAge);
        buttonSubmit = findViewById(R.id.buttonSubmit);

        buttonSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // Handle submit button click, you can retrieve values from the fields here
                String firstName = editTextFirstName.getText().toString();
                String lastName = editTextLastName.getText().toString();
                String email = editTextEmail.getText().toString();
                int age = Integer.parseInt(editTextAge.getText().toString());

                // Display a greeting message using a Toast
                String greetingMessage = "Hello, " + firstName + " " + lastName + "!\n" +
                        "Email: " + email + "\n";

                Toast.makeText(MainActivity.this, greetingMessage, Toast.LENGTH_LONG).show();
```

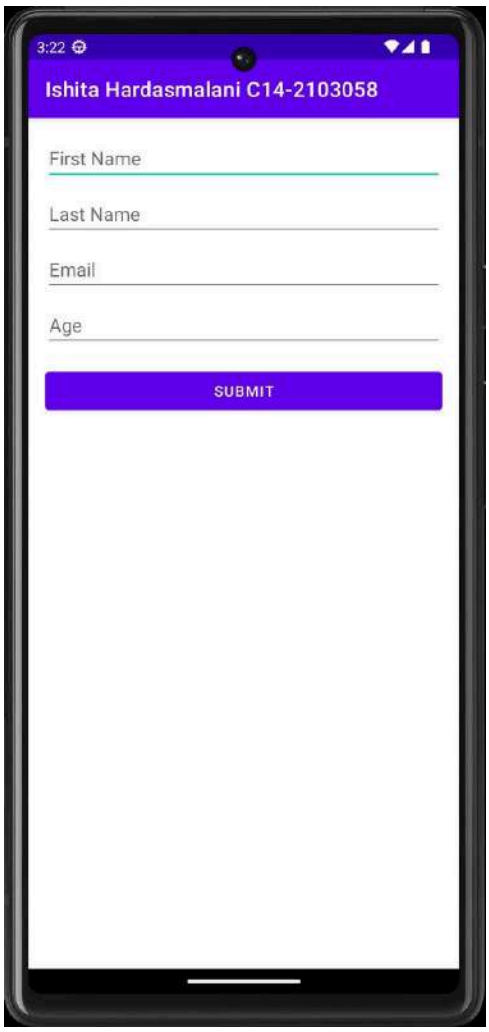
```
        }
      });
    }
}
```

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns: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"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editTextFirstName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="First Name"/>

    <EditText
        android:id="@+id/editTextLastName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Last Name"
        android:layout_marginTop="8dp"/>

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:layout_marginTop="8dp"/>

    <EditText
        android:id="@+id/editTextAge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Age"
        android:inputType="number"
```

```
        android:layout_marginTop="8dp"/>

    <Button
        android:id="@+id/buttonSubmit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:layout_marginTop="16dp"/>
</LinearLayout>
```

OUTPUT:



THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

EXPERIMENT NO. 4.

(13)

Aim: write an android application to design GUI component using database

Theory:

The Android application under consideration is a user registration & display system designed to capture user information, store it in a sqlite database & subsequently display the stored user details in a Recycler view.

Components:

Main Activity.java:

→ The entry point of the application. Main Activity is responsible for capturing user details such as name, email, contact number & gender.
User input is validated & if successful, the data is inserted into an sqlite database using 'DatabaseHelper'.

Functions used:

→ validateForm(): Validates user input before inserting into the database. Displays appropriate error messages using TextInputLayout.setError() & Toast messages for missing / incorrect inputs.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

→ btnSignUp.setOnClickListener():
After validation, retrieves user input & inserts into the database using databaseHelper.insertUserDetails(). Displays success/failure toast messages. Finally, restarts the MainActivity to reset the form.

→ btnViewUsers.setOnClickListener():
Implements a click listener for 'view users' button. Navigates to the secondActivity using an Intent.

SecondActivity.java: Retrieves user details from sqlite database. If no users are found, a message is displayed indicating absence of user details. Otherwise, RecyclerView is populated with user information.

DatabaseHelper.java.
Class responsible for creating & managing the users table in the SQLite database. Executes an sqlite command to create 'users' table with columns: id, Name, email, Contact no & Gender. CRUDs is done with onCreate function

UserRegularViewAdapter.java: An adapter class that connects the data source to the RecyclerView in secondActivity
Methods used: onCreateViewHolder(), onBindViewHolder(), getItemCount(), UserRecyclerViewHolder().

EXPERIMENT 04

CODE:

MainActivity.java

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.RadioGroup;
import android.widget.ScrollView;
import android.widget.Toast;
import com.google.android.material.textfield.TextInputEditText;
import com.google.android.material.textfield.TextInputLayout;
public class MainActivity extends AppCompatActivity {
    Context context;
    DatabaseHelper databaseHelper;
    ScrollView svRegistrationForm;
    TextInputLayout tilName, tilEmail, tilContactNo;
    TextInputEditText tietName, tietEmail, tietContactNo;
    RadioGroup rgGender;
    Button btnSignUp, btnViewUsers;
    String name, email, contactNo, gender;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        context = MainActivity.this;
        databaseHelper = new DatabaseHelper(context);
        svRegistrationForm = findViewById(R.id.sv_registration_form);
        tilName = findViewById(R.id.til_name);
        tilEmail = findViewById(R.id.til_email);
        tilContactNo = findViewById(R.id.til_contact_no);
        tietName = findViewById(R.id.tiet_name);
        tietEmail = findViewById(R.id.tiet_email);
        tietContactNo = findViewById(R.id.tiet_contact_no);
        rgGender = findViewById(R.id.rg_gender);
        btnSignUp = findViewById(R.id.btn_insert_data);
        btnViewUsers = findViewById(R.id.btn_view_users);
        rgGender.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
```

```
                                        @Override
                                        public void onCheckedChanged(RadioGroup group, int
checkedId) {
                                            if(checkedId == R.id.rb_male) {
                                                gender = "Male";
                                            } else {
                                                gender = "Female";
                                            }
                                        }
                                    });
        btnSignUp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(validateForm()) {
                    name = tietName.getText().toString();
                    email = tietEmail.getText().toString();
                    contactNo = tietContactNo.getText().toString();
                    boolean insertResult = databaseHelper.insertUserDetails(email, name,
                            contactNo,
                            gender);
                    if(insertResult) {
                        Toast.makeText(context, "Data inserted succesfully",
                                Toast.LENGTH_SHORT).show();
                    } else {
                        Toast.makeText(context, "Data not inserted",
                                Toast.LENGTH_SHORT).show();
                    }
                    Intent intent = new Intent(context, MainActivity.class);
                    startActivity(intent);
                }
            }
        });
        btnViewUsers.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(context, SecondActivity.class);
                startActivity(intent);
            }
        });
    }
    public boolean validateForm() {
        if(tietName.getText().toString().isEmpty()) {
            tilName.setError("Enter Name");
            focusOnView(tilName);
```

```
            return false;
        } else if(tietEmail.getText().toString().isEmpty()) {
            tilEmail.setError("Enter Email");
            focusOnView(tilEmail);
            return false;
        } else if(tietContactNo.getText().toString().isEmpty()) {
            tilContactNo.setError("Enter Contact no");
            focusOnView(tilContactNo);
            return false;
        } else if(rgGender.getCheckedRadioButtonId() == -1) {
            Toast.makeText(context, "Select Gender", Toast.LENGTH_SHORT).show();
            focusOnView(rgGender);
            return false;
        } else {
            return true;
        }
    }
    public void focusOnView(final View view) {
        svRegistrationForm.post(new Runnable() {
            @Override
            public void run() {
                svRegistrationForm.smoothScrollTo(0, view.getTop()-50);
            }
        });
    }
}
```

SecondActivity.java

```
package com.example.myapplication;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.ListView;
import android.widget.TextView;
import java.util.ArrayList;
public class SecondActivity extends AppCompatActivity {
    Context context;
```

```
    DatabaseHelper databaseHelper;
    ArrayList<ModelForUsers> userList = new ArrayList();
    RecyclerView rvUserDetails;
    RecyclerView.Adapter adapter;
    TextView tvMessage;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        context = SecondActivity.this;
        databaseHelper = new DatabaseHelper(context);
        rvUserDetails = findViewById(R.id.rv_user_details);
        tvMessage = findViewById(R.id.tv_message);
        Cursor cursor = databaseHelper.getAllUserDetails();
        if(cursor.getCount() == 0) {
            tvMessage.setVisibility(View.VISIBLE);
        } else {
            tvMessage.setVisibility(View.GONE);
            while (cursor.moveToNext()) {
                ModelForUsers user = new ModelForUsers();
                user.setId(cursor.getString(0));
                user.setName(cursor.getString(1));
                user.setEmail(cursor.getString(2));
                user.setContactNo(cursor.getString(3));
                user.setGender(cursor.getString(4));
                userList.add(user);
            }
            rvUserDetails.setHasFixedSize(true);
            rvUserDetails.setLayoutManager(new LinearLayoutManager(this));
            adapter = new UserRecyclerViewAdapter(context, userList);
            rvUserDetails.setAdapter(adapter);
        }
    }
}
```

DatabaseHelper.java

```
package com.example.myapplication;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.MatrixCursor;
import android.database.SQLException;
```

```
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
import androidx.annotation.Nullable;
import java.util.ArrayList;
public class DatabaseHelper extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "User Data DB";
    public static final String TABLE_NAME = "Users";
    // COL_0 will be user Id which is AUTOINCREMENT
    public static final String COL_1 = "Email";
    public static final String COL_2 = "Name";
    public static final String COL_3= "Contact_No";
    public static final String COL_4 = "Gender";
    public DatabaseHelper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create table Users (Id INTEGER PRIMARY KEY AUTOINCREMENT,Name
TEXT, Email TEXT,Contact_No TEXT, Gender TEXT)");
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS Users");
        onCreate(db);
    }
    // method for inserting data of a single user
    public boolean insertUserDetails(String email, String name, String contactNo, String gender) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(COL_1, email);
        contentValues.put(COL_2, name);
        contentValues.put(COL_3, contactNo);
        contentValues.put(COL_4, gender);
        long insertResult = db.insert(TABLE_NAME, null, contentValues);
        if(insertResult == -1) {
            return false;
        } else {
            return true;
        }
    }
    // method for get data of all users
    public Cursor getAllUserDetails() {
        SQLiteDatabase db = this.getWritableDatabase();
```

```
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
        return cursor;
    }
}
```

ModelForUsers.java

```
package com.example.myapplication;

public class ModelForUsers {
    String id, name, email, contactNo, gender;
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getContactNo() {
        return contactNo;
    }
    public void setContactNo(String contactNo) {
        this.contactNo = contactNo;
    }
    public String getGender() {
        return gender;
    }
    public void setGender(String gender) {
        this.gender = gender;
    }
}
```

UserRecyclerViewAdapter.java

```
package com.example.myapplication;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
public class UserRecyclerViewAdapter extends
      RecyclerView.Adapter<UserRecyclerViewAdapter.UserRecyclerViewHolder> {
   Context context;
   ArrayList<ModelForUsers> usersList;
   public UserRecyclerViewAdapter(Context context, ArrayList<ModelForUsers>
         usersList) {
      this.context = context;
      this.usersList = usersList;
   }
   @NonNull
   @Override
   public UserRecyclerViewHolder onCreateViewHolder(@NonNull ViewGroup
                                          parent, int viewType) {
      LayoutInflater layoutInflater = LayoutInflater.from(parent.getContext());
      View view = layoutInflater.inflate(R.layout.row_users, parent, false);
      UserRecyclerViewHolder userRecyclerViewHolder = new
            UserRecyclerViewHolder(view);
      return userRecyclerViewHolder;
   }
   @Override
   public void onBindViewHolder(@NonNull UserRecyclerViewHolder
                                userRecyclerViewHolder, int position) {
      userRecyclerViewHolder.tvId.setText("Id: " + usersList.get(position).getId());
      userRecyclerViewHolder.tvName.setText("Name: " +
            usersList.get(position).getName());
      userRecyclerViewHolder.tvEmail.setText("Email: " +
            usersList.get(position).getEmail());
      userRecyclerViewHolder.tvContactNo.setText("Contact No: " +
            usersList.get(position).getContactNo());
      userRecyclerViewHolder.tvGender.setText("Gender: " +
            usersList.get(position).getGender());
   }
   @Override
```

```
public int getItemCount() {
    return usersList.size();
}
class UserRecyclerViewHolder extends RecyclerView.ViewHolder {
    TextView tvId, tvName, tvEmail, tvContactNo, tvGender;
    public UserRecyclerViewHolder(@NonNull View itemView) {
        super(itemView);
        tvId = itemView.findViewById(R.id.tv_id);
        tvName = itemView.findViewById(R.id.tv_name);
        tvEmail = itemView.findViewById(R.id.tv_email);
        tvContactNo = itemView.findViewById(R.id.tv_contact_no);
        tvGender = itemView.findViewById(R.id.tv_gender);
    }}}
```

XML:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <ScrollView
        android:id="@+id/sv_registration_form"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="10dp"
            android:orientation="vertical">
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/til_name"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
```

```
    android:layout_marginTop="20dp">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/tiet_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/til_email"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="20dp">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/tiet_email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/til_contact_no"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="20dp">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/tiet_contact_no"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Contact No."
        android:inputType="number"
        android:maxLength="10"/>
</com.google.android.material.textfield.TextInputLayout>
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Gender"
    android:textSize="18sp"
    android:layout_marginTop="20dp"/>
<RadioGroup
    android:id="@+id/rg_gender"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <RadioButton
        android:id="@+id/rb_male"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Male"
            android:textSize="15sp"/>
        <RadioButton
            android:id="@+id/rb_female"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Female"
            android:textSize="15sp"/>
    </RadioGroup>
    <Button
        android:id="@+id/btn_insert_data"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Insert Data"
        android:textAllCaps="false"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"/>
    <Button
        android:id="@+id/btn_view_users"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="View Users"
        android:textAllCaps="false"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"/>
  </LinearLayout>
 </ScrollView>
</LinearLayout>
```

activity_second.xml

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

```
        android:id="@+id/tv_message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#ffffff"
        android:layout_margin="20dp"
        android:text="No user details"
        android:textSize="30dp"
        android:gravity="center"
        android:visibility="gone"/>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_margin="20dp">
        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/rv_user_details"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
    </LinearLayout>
</LinearLayout>
```

row_users.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="#ffffff"
    android:padding="10dp">
    <TextView
        android:id="@+id/tv_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textColor="#000000"
        android:text="id"/>
    <TextView
        android:id="@+id/tv_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
```

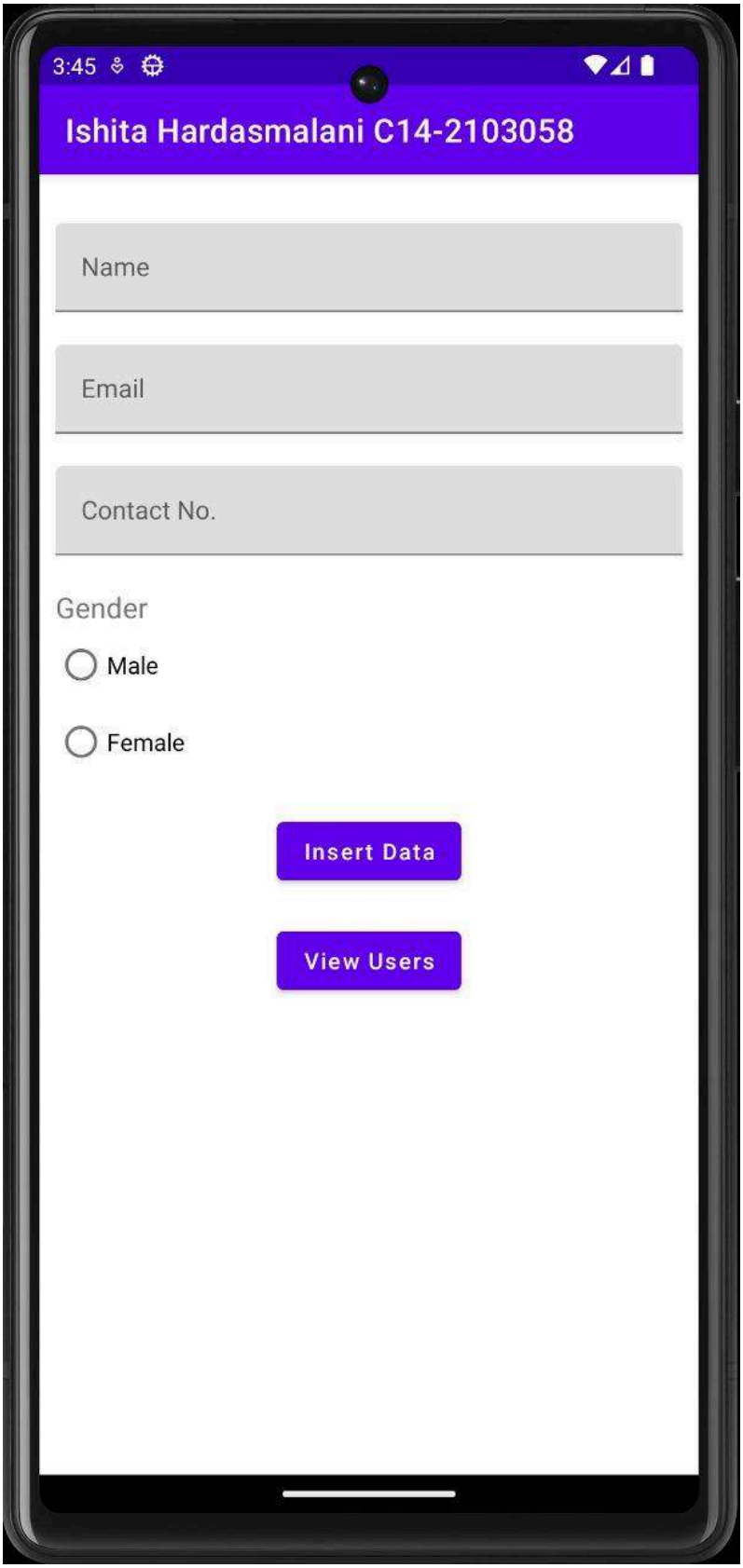
```
        android:textColor="#000000"
        android:text="name"/>
    <TextView
        android:id="@+id/tv_email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textColor="#000000"
        android:layout_marginTop="5dp"
        android:text="email"/>
    <TextView
        android:id="@+id/tv_contact_no"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textColor="#000000"
        android:layout_marginTop="5dp"
        android:text="contact"/>
    <TextView
        android:id="@+id/tv_gender"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textColor="#000000"
        android:layout_marginTop="5dp"
        android:text="gender"/>
</LinearLayout>
```

AndroidManifest.xml

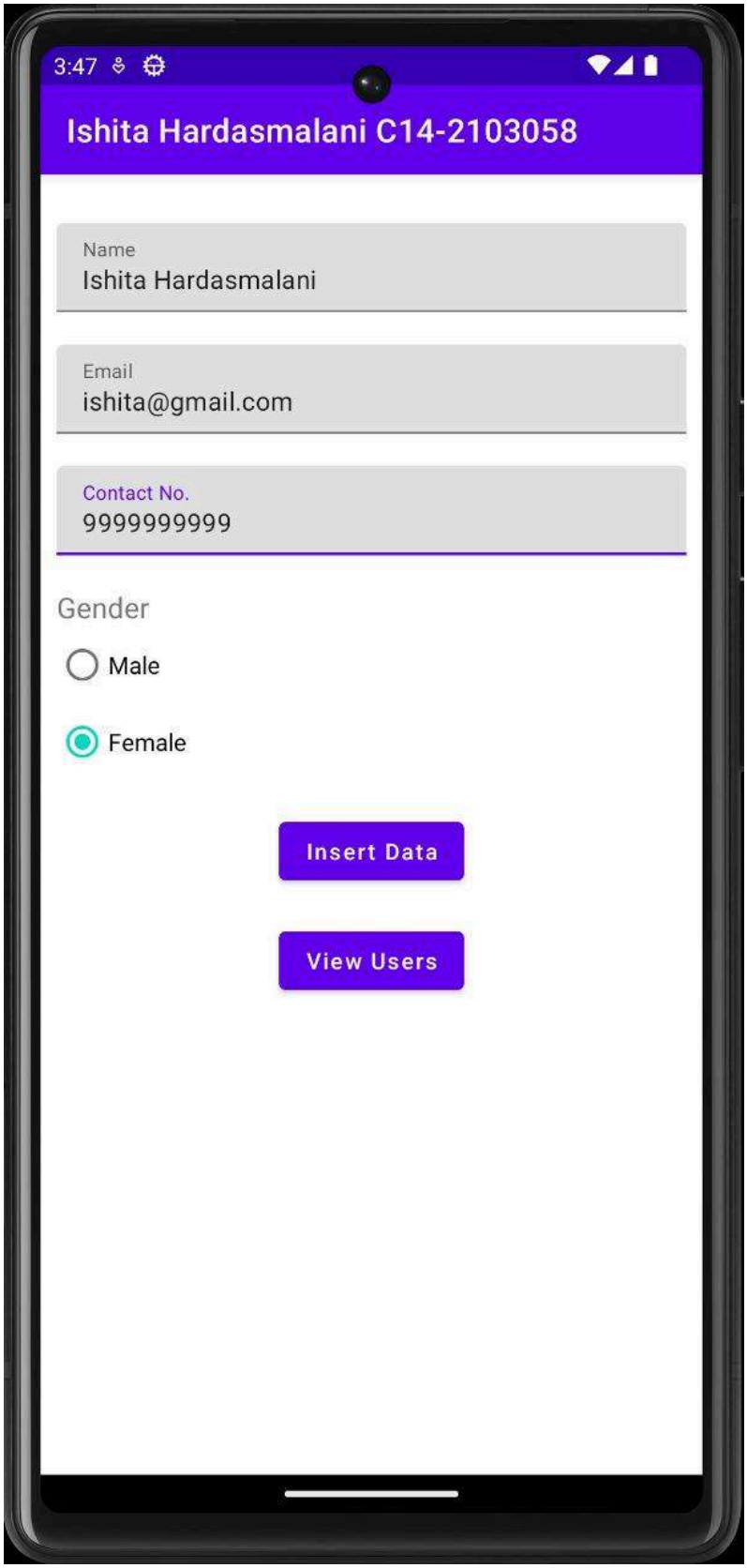
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication"
        tools:targetApi="31" >
```

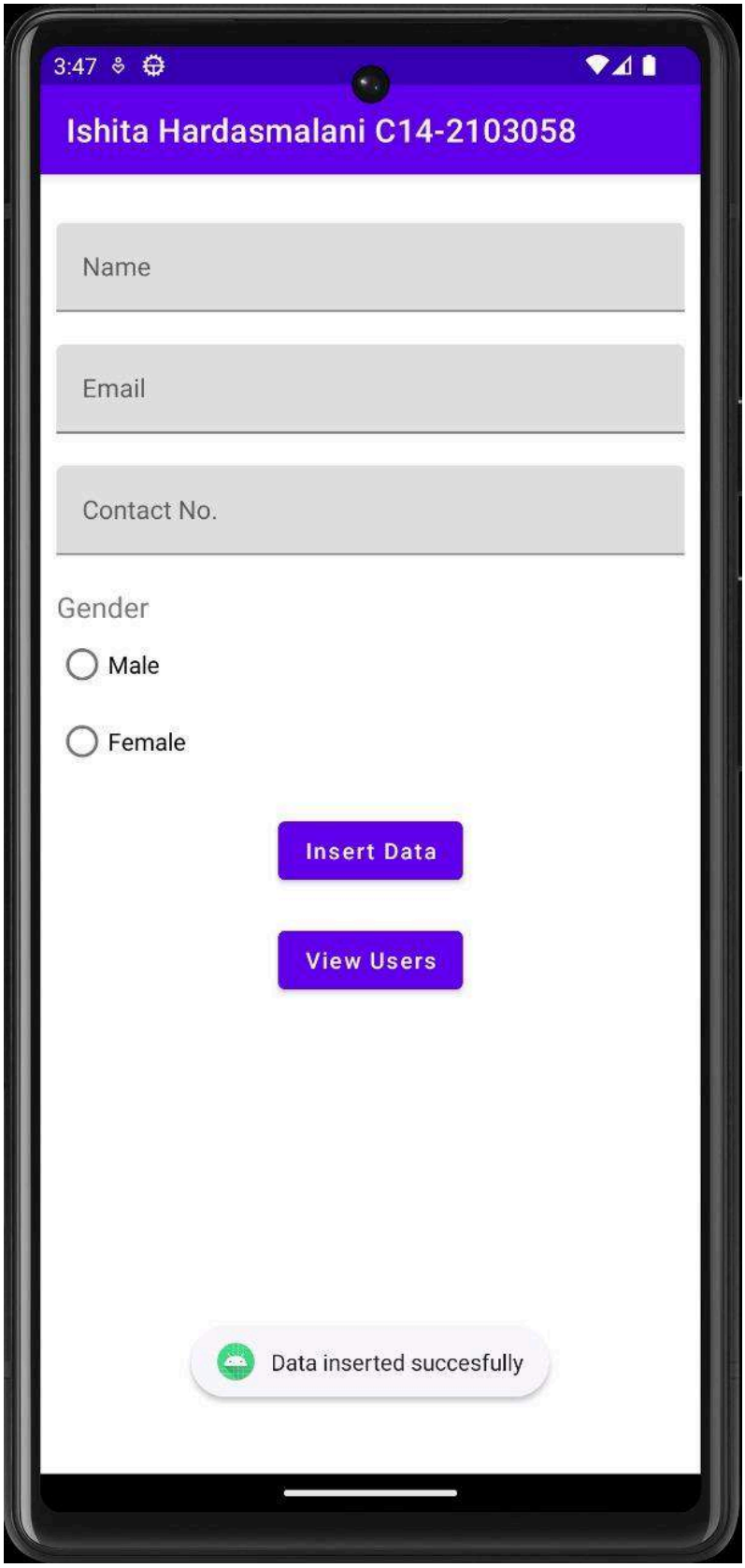
```
        <activity android:exported="true" android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:exported="true" android:name=".SecondActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>



    </application>
</manifest>
```

OUTPUT:


3:45
Ishita Hardasmalani C14-2103058
Name
Email
Contact No.
Gender
Male
Female
Insert Data
View Users


3:45
Ishita Hardasmalani C14-2103058
No user details


3:47
Ishita Hardasmalani C14-2103058
Name
Ishita Hardasmalani
Email
ishita@gmail.com
Contact No.
9999999999
Gender
Male
Female
Insert Data
View Users


3:47
Ishita Hardasmalani C14-2103058
Name
Email
Contact No.
Gender
Male
Female
Insert Data
View Users
Data inserted succesfully


3:47
Ishita Hardasmalani C14-2103058
Id: 1
Name: Ishita Hardasmalani
Email: ishita@gmail.com
Contact No: 9999999999
Gender: Female

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

(A)

EXPERIMENT NO. 5

Aim: Write an Android application to develop an EMI calculator application.

Theory: This experiment is for an android application developed with the android SDK, specifically aimed at calculating the Equated Monthly Installment (EMI) for a loan.

Methods used.

① emiCalcBtn.setOnClickListener():
Implements a click listener for "calculate" button. Retrieves the user inputs for principal (P), Interest rate (I) & Tenure (T).
Validates the input using TextUtils.isEmpty() to ensure the fields principal amount, interest rate & tenure are provided. Parses the i/p strings into float values & performs the necessary calculations. Calls a series of calculation methods to compute the EMI.
Sets the calculated EMI value in the 'EditText' field for EMI ('result').

Calculation Methods:

→ calPric (float p):
Returns principal amount.

→ calInt (float i):
Returns monthly interest rate.

→ calMonth (float y):
Returns total no. of monthly installments

→ calDvdnt (float Rate, float Months):
Returns the value of '(1+Rate)^Months'.

→ calFinalDvdnt ():
Returns the final dividend for the EMI calculation.

→ calDivider (float Dvdnt):
Returns the value of '(Dvdnt - 1)'.

→ calEMI (float FD, float D):
Returns the calculated EMI using the formula '(FD/D)'.

EXPERIMENT - 05

CODE:

MainActivity.java

```
package com.example.myapplication;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    Button emiCalcBtn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final EditText P = (EditText)findViewById(R.id.principal);
        final EditText I = (EditText)findViewById(R.id.interest);
        final EditText Y = (EditText)findViewById(R.id.years);
        final EditText result = (EditText)findViewById(R.id.emi);
        emiCalcBtn = (Button) findViewById(R.id.btn_calculate2);
        emiCalcBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String st1 = P.getText().toString();
                String st2 = I.getText().toString();
                String st3 = Y.getText().toString();
                if (TextUtils.isEmpty(st1)) {
                    P.setError("Enter Prncipal Amount");
                    P.requestFocus();
                    return;
                }
                if (TextUtils.isEmpty(st2)) {
                    I.setError("Enter Interest Rate");
                    I.requestFocus();
                    return;
                }
                if (TextUtils.isEmpty(st3)) {
                    Y.setError("Enter Years");
                    Y.requestFocus();
                    return;
```

```
            }
            float p = Float.parseFloat(st1);
            float i = Float.parseFloat(st2);
            float y = Float.parseFloat(st3);
            float Principal = calPric(p);
            float Rate = calInt(i);
            float Months = calMonth(y);
            float Dvdnt = calDvdnt( Rate, Months);
            float FD = calFinalDvdnt (Principal, Rate, Dvdnt);
            float D = calDivider(Dvdnt);
            float emi = calEmi(FD, D);
            result.setText(String.valueOf(emi));
        }
      });
  }
  public float calPric(float p) {
     return (float) (p);
  }
  public float calInt(float i) {
     return (float) (i/12/100);
  }
  public float calMonth(float y) {
     return (float) (y * 12);
  }
  public float calDvdnt(float Rate, float Months) {
     return (float) (Math.pow(1+Rate, Months));
  }
  public float calFinalDvdnt(float Principal, float Rate, float Dvdnt) {
     return (float) (Principal * Rate * Dvdnt);
  }
  public float calDivider(float Dvdnt) {
     return (float) (Dvdnt-1);
  }
  public float calEmi(float FD, Float D) {
     return (float) (FD/D);
  }
}
```

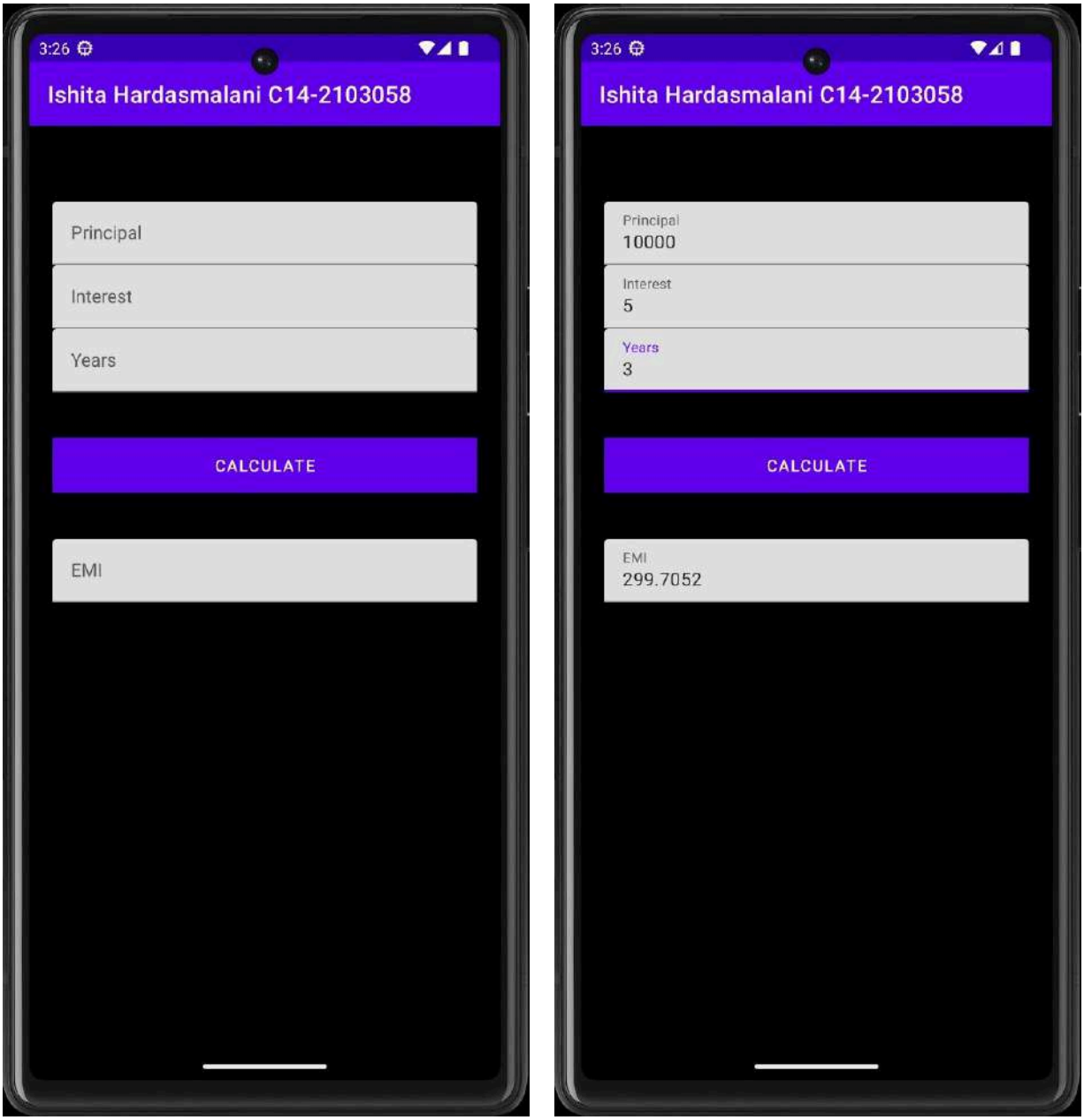
XML:

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   xmlns:tools="http://schemas.android.com/tools"
   xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
tools:context="com.example.myapplication.MainActivity"
android:layout_height="match_parent"
android:background="@color/black"
android:backgroundTint="@color/black">
<androidx.core.widget.NestedScrollView
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   app:layout_behavior="@string/appbar_scrolling_view_behavior">
   <LinearLayout
      android:layout_width="fill_parent"
      android:layout_height="match_parent"
      android:layout_marginTop="?attr/actionBarSize"
      android:orientation="vertical"
      android:paddingLeft="20dp"
      android:paddingRight="20dp"
      android:paddingTop="10dp">
      <com.google.android.material.textfield.TextInputLayout
         android:id="@+id/input_layout_principal"
         android:layout_width="match_parent"
         android:layout_height="wrap_content">
         <EditText
            android:id="@+id/principal"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:singleLine="true"
            android:inputType="number"
            android:digits="0123456789."
            android:hint="Principal" />
      </com.google.android.material.textfield.TextInputLayout>
      <com.google.android.material.textfield.TextInputLayout
         android:id="@+id/input_layout_interest"
         android:layout_width="match_parent"
         android:layout_height="wrap_content">
         32
         <EditText android:id="@+id/interest"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:singleLine="true"
            android:inputType="number"
            android:digits="0123456789."
            android:hint="Interest" />
      </com.google.android.material.textfield.TextInputLayout>
      <com.google.android.material.textfield.TextInputLayout
         android:id="@+id/input_layout_tenure"
```

```
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <EditText
                android:id="@+id/years"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:inputType="number"
                android:digits="0123456789."
                android:hint="Years" />
        </com.google.android.material.textfield.TextInputLayout>
        <Button android:id="@+id/btn_calculate2"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Calculate"
            android:background="#000000"
            android:layout_marginTop="40dp"
            android:textColor="#FFFFFF"/>
        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/input_layout_emi"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="40dp">
            <EditText android:id="@+id/emi"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:maxEms="0"
                android:inputType="number"
                android:hint="EMI" />
        </com.google.android.material.textfield.TextInputLayout>
    </LinearLayout>
  </androidx.core.widget.NestedScrollView>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

OUTPUT:


3:26
Ishita Hardasmalani C14-2103058
Principal
Interest
Years
CALCULATE
EMI
3:26
Ishita Hardasmalani C14-2103058
Principal
10000
Interest
5
Years
3
CALCULATE
EMI
299.7052

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

EXPERIMENT NO. 6

Aim: Write an android application that creates an alert on receiving a message.

Theory:
This experiment demonstrates the implementation of a simple Android application that generates notifications. In android, the notification system is managed by several classes & packages, including 'Notification Manager', 'Notification compat. Builder', 'Pending Intent' & 'Notification channel'.

- Overview of the Notification process:

1. Create Notification channel (for Android 8.0 & above).
Instantiate a 'Notification channel' object.
Set channel attributes such as name, description, importance & enable features like lights & vibrations. This is done with methods: setName(), setDescription(), setImportance(), enableLights.
Register the channel with the 'Notification Manager'.

2. Build Notification:
Instantiate a 'Notification. Builder' object.
Set various attributes of the notification, including title, icon, content & intent

Using setContentTitle(), setContentText(), setSmallIcon(), setContentIntent() methods.
Call 'builder' to get the final 'Notification' object.

3. Show Notification:
Obtain an instance of 'NotificationManager'.
Use 'notify(int id, Notification notification)' to display notification from object obtained in previous step.

4. Handle Notification click:
Specify an 'PendingIntent' that will be triggered when the user clicks on the notifications. This 'PendingIntent' often leads to an activity or service within your app.

EXPERIMENT - 06

CODE:

```
package com.example.myapplication;

import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;

public class MainActivity extends AppCompatActivity {
    Button btnNotify;
    EditText etMessage;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnNotify = findViewById(R.id.btn_notify);
        etMessage = findViewById(R.id.et_message);

        btnNotify.setOnClickListener(new View.OnClickListener() {
            @RequiresApi(api = Build.VERSION_CODES.O)
            @Override
            public void onClick(View v) {
                String message = etMessage.getText().toString();
                Intent intent = new Intent(MainActivity.this, MainActivity.class);
                PendingIntent pendingIntent = PendingIntent.getActivity(
                        MainActivity.this, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT |
PendingIntent.FLAG_MUTABLE);

                NotificationManager mNotificationManager = (NotificationManager)
```

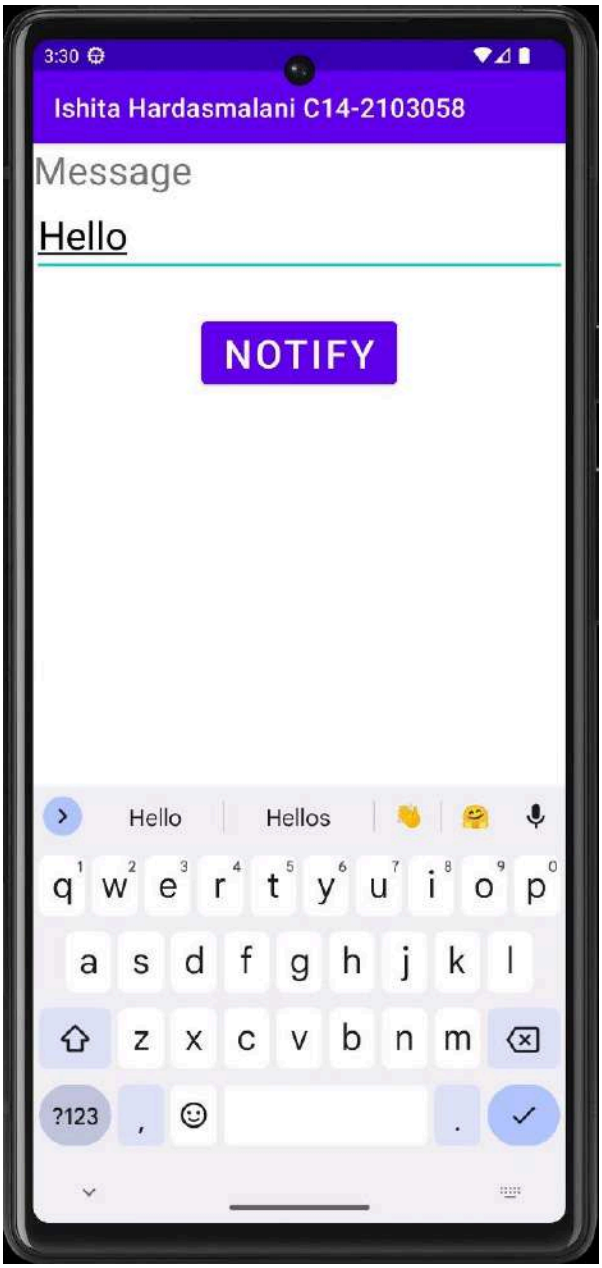
```
                getSystemService(Context.NOTIFICATION_SERVICE);

            String id = "243";
            String name = "My Channel"; // Set a meaningful name for your channel
            int importance = NotificationManager.IMPORTANCE_LOW;

            NotificationChannel mChannel = new NotificationChannel(id, name, importance);
            mChannel.enableLights(true);
            mNotificationManager.createNotificationChannel(mChannel);

            NotificationCompat.Builder notificationBuilder = new
NotificationCompat.Builder(MainActivity.this, id)
                    .setContentTitle("New Message")
                    .setContentText(etMessage.getText().toString())
                    .setSmallIcon(R.mipmap.ic_launcher)
                    .setContentIntent(pendingIntent);

            Notification notification = notificationBuilder.build();

            notification.flags |= Notification.FLAG_AUTO_CANCEL;
            mNotificationManager.notify(Integer.parseInt(id), notification);

            etMessage.setText("");
        }
    });
  }
}
```

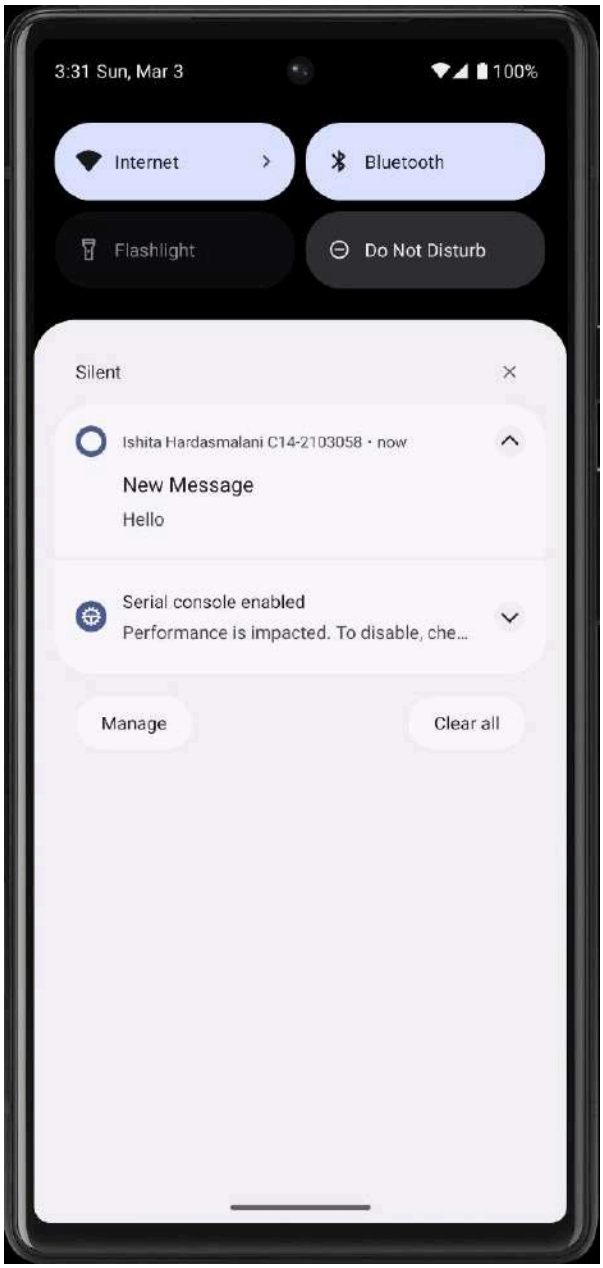
XML

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

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Message"
        android:textSize="30sp" />

    <EditText
        android:id="@+id/et_message"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:singleLine="true"
        android:textColor="#000000"
        android:textSize="30sp" />

    <Button
        android:id="@+id/btn_notify"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="30dp"
        android:text="Notify"
        android:textSize="30sp" />

</LinearLayout>
```

OUTPUT


3:30
Ishita Hardasmalani C14-2103058
Message
Hello
NOTIFY
Hello
Hellos
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123


3:31 Sun, Mar 3
100%
Internet
Bluetooth
Flashlight
Do Not Disturb
Silent
Ishita Hardasmalani C14-2103058 · now
New Message
Hello
Serial console enabled
Performance is impacted. To disable, che...
Manage
Clear all

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

EXPERIMENT NO. 7.

A+

Aim: Write an android application to implement Basic calculator.

Theory.

- activity_main.xml:
  Defines a vertical 'linear layout' to organize UI elements. Two 'EditText' fields for entering numbers. A 'TextView' for displaying the result. Buttons for basic math(calculator operations & a clear button.

- Main Activity:
  * Initialization & setup.
    'etFirstNumber', 'etSecondNumber' : instances of 'EditText' for input of first & second number.
    'tvResult' : Instance of 'TextView' to display result.
    'btnAdd', 'btnSubtract', 'btnMultiply', 'btnDivide' 'btnClear' : Instances of 'Button' for different operations & clearing.

- onCreate():
  Sets content view to XML layout. Retrieves UI using 'findViewById'. Setsup click listeners for each operation button (+, -, /, *) & the clear button.

- performOperation():
  takes an operator as an argument.
  checks if both 'EditText' fields are not empty. Parses the input numbers.
  Performs specified operation & displays result to 'TextView'.

- clearFields():
  clears the content of both 'EditText' fields & the 'TextView'.

EXPERIMENT - 07

CODE:

```
package com.example.myapplication;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    EditText etFirstNumber, etSecondNumber;
    TextView tvResult;
    Button btnAdd, btnSubtract, btnMultiply, btnDivide, btnClear;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        etFirstNumber = findViewById(R.id.et_first_number);
        etSecondNumber = findViewById(R.id.et_second_number);
        tvResult = findViewById(R.id.tv_result);
        btnAdd = findViewById(R.id.btn_add);
        btnSubtract = findViewById(R.id.btn_subtract);
        btnMultiply = findViewById(R.id.btn_multiply);
        btnDivide = findViewById(R.id.btn_divide);
        btnClear = findViewById(R.id.btn_clear);

        btnAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performOperation("+");
            }
        });

        btnSubtract.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performOperation("-");
            }
        });
```

```
        btnMultiply.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performOperation("*");
            }
        });

        btnDivide.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                performOperation("/");
            }
        });

        btnClear.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                clearFields();
            }
        });
    }

    private void performOperation(String operator) {
        if (etFirstNumber.getText().toString().isEmpty() ||
etSecondNumber.getText().toString().isEmpty()) {
            tvResult.setText("Enter numbers first");
            return;
        }

        int firstNumber = Integer.parseInt(etFirstNumber.getText().toString());
        int secondNumber = Integer.parseInt(etSecondNumber.getText().toString());
        int result;

        switch (operator) {
            case "+":
                result = firstNumber + secondNumber;
                tvResult.setText("Addition of " + firstNumber + " + " + secondNumber + " is " + result);
                break;
            case "-":
                result = firstNumber - secondNumber;
                tvResult.setText("Subtraction of " + firstNumber + " - " + secondNumber + " is " +
result);
                break;
```

```
            case "*":
                result = firstNumber * secondNumber;
                tvResult.setText("Multiplication of " + firstNumber + " * " + secondNumber + " is " +
result);
                break;
            case "/":
                if (secondNumber == 0) {
                    tvResult.setText("Cannot divide by zero");
                    return;
                }
                result = firstNumber / secondNumber;
                tvResult.setText("Division of " + firstNumber + " / " + secondNumber + " is " + result);
                break;
        }
    }

    private void clearFields() {
        etFirstNumber.setText("");
        etSecondNumber.setText("");
        tvResult.setText("");
    }
}
```

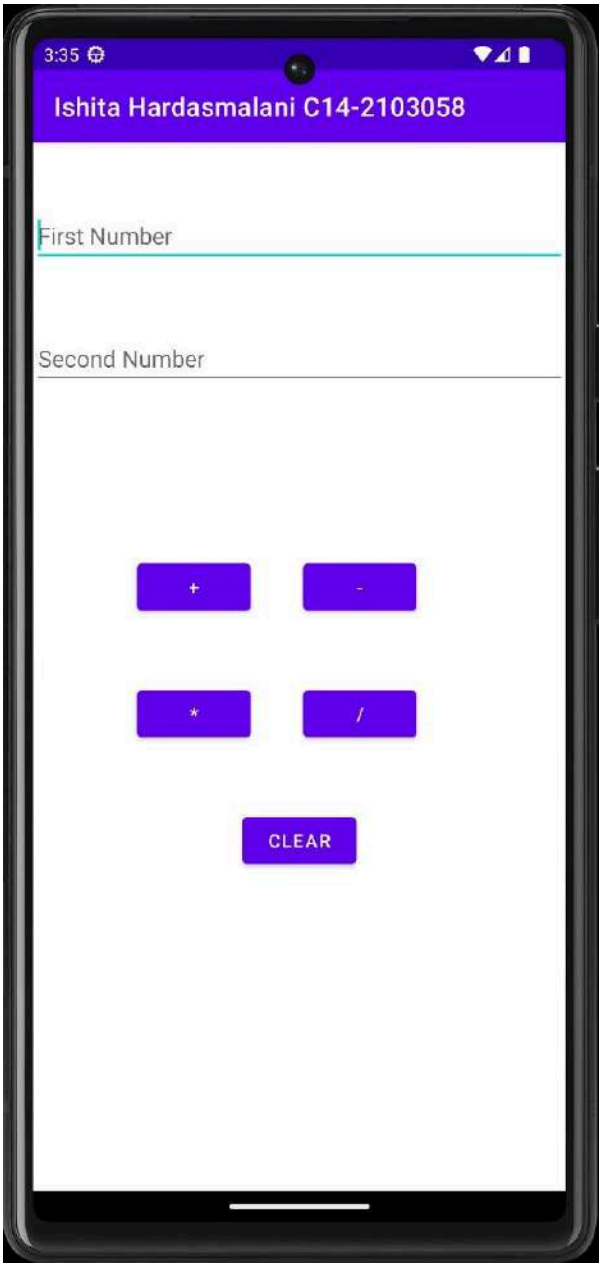
XML:

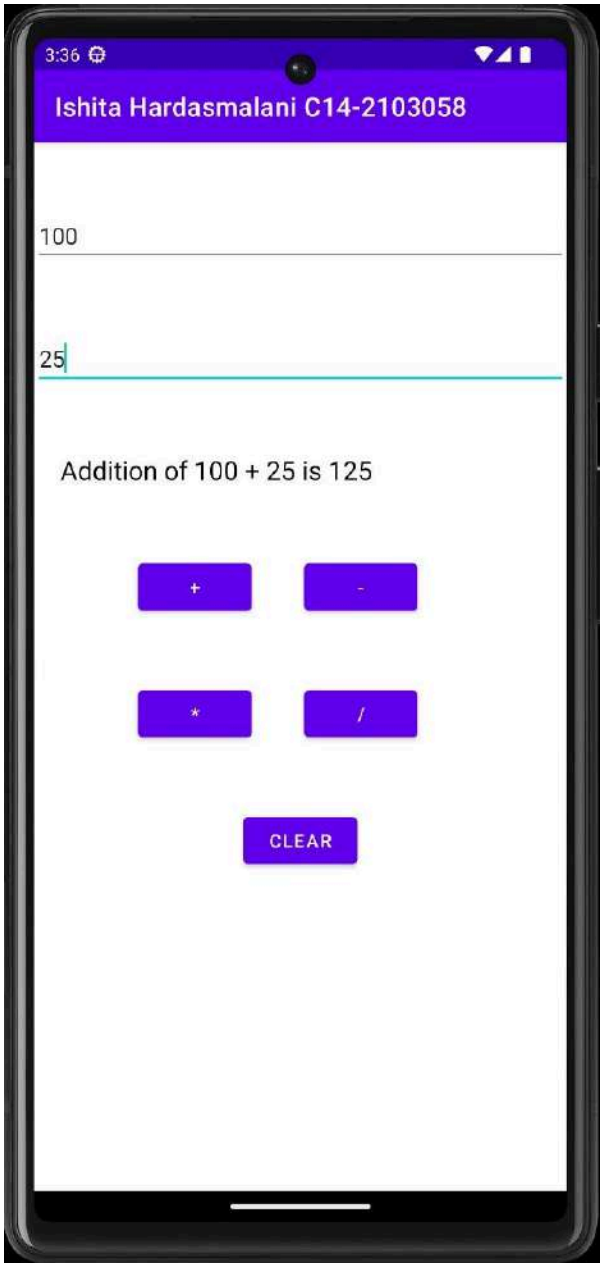
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <EditText
        android:id="@+id/et_first_number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="number"
        android:layout_marginTop="50dp"
        android:hint="First Number"/>

    <EditText
        android:id="@+id/et_second_number"
```

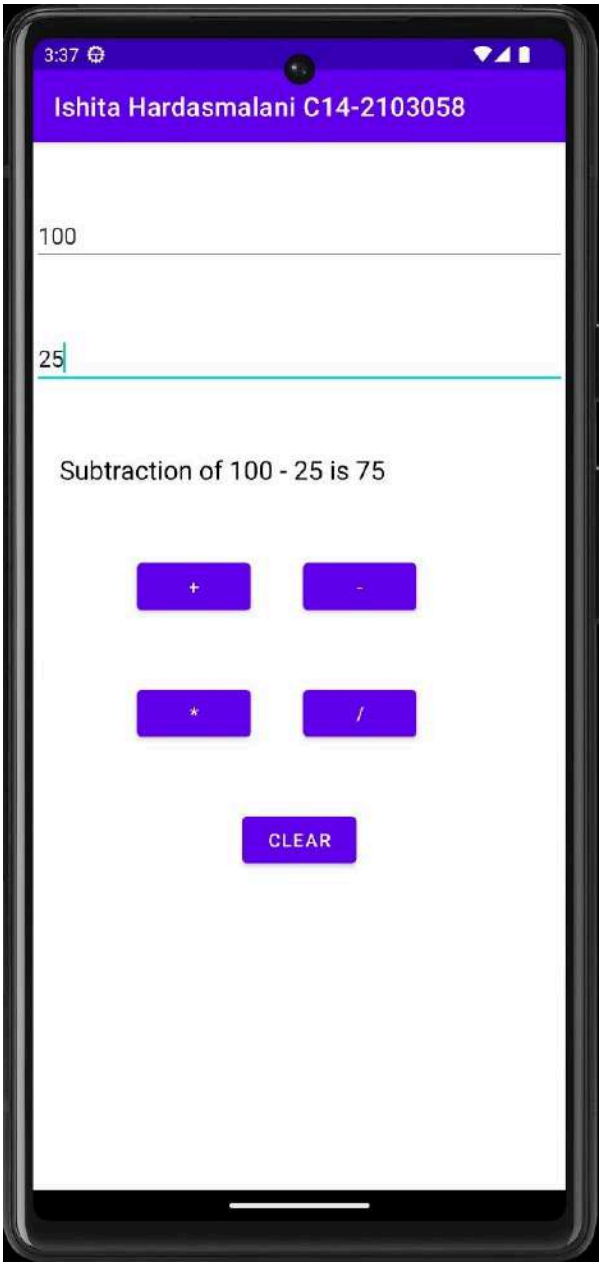
```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="number"
    android:layout_marginTop="50dp"
    android:hint="Second Number"/>

<TextView
    android:id="@+id/tv_result"
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:textSize="20sp"
    android:textColor="#000000"
    android:text=" "
    android:layout_marginTop="50dp"
    android:paddingLeft="20dp"/>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginLeft="50dp">

    <Button
        android:id="@+id/btn_add"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="+"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="20dp"/>

    <Button
        android:id="@+id/btn_subtract"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="-"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

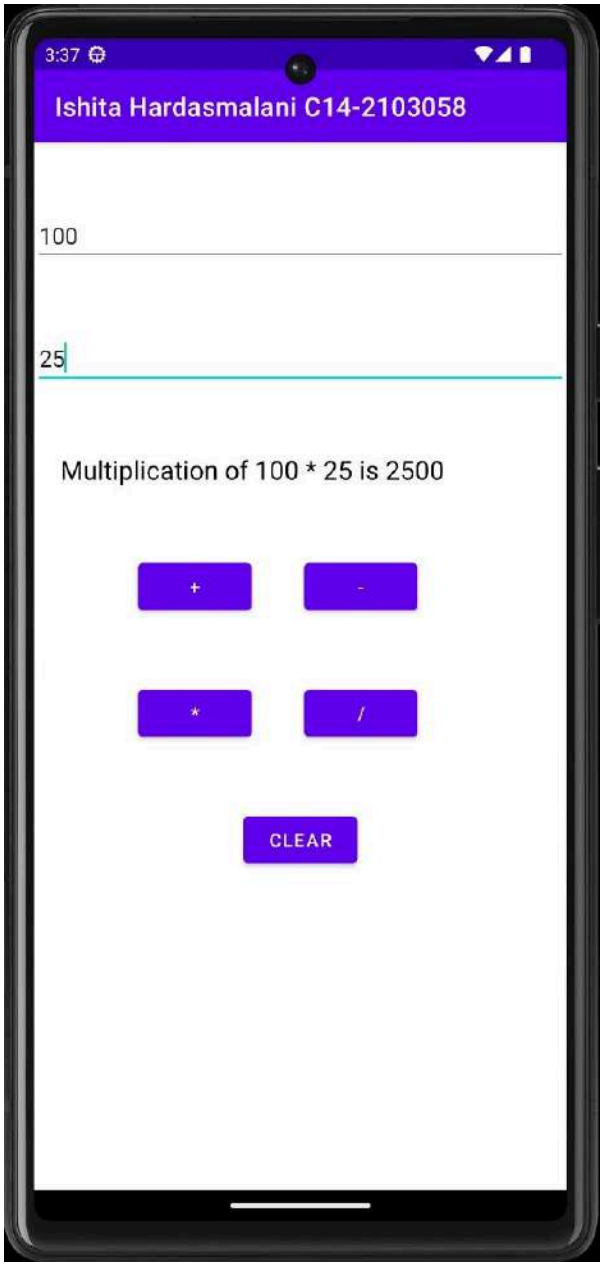
```
        android:layout_marginTop="50dp"
        android:layout_marginRight="50dp"
        android:layout_marginLeft="50dp">

        <Button
            android:id="@+id/btn_multiply"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="*"
            android:layout_marginLeft="30dp"
            android:layout_marginRight="20dp"/>

        <Button
            android:id="@+id/btn_divide"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="/"
            android:layout_marginRight="20dp"
            android:layout_marginLeft="20dp"/>
    </LinearLayout>

    <Button
        android:id="@+id/btn_clear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Clear"
        android:layout_marginTop="50dp"
        android:layout_gravity="center"/>

</LinearLayout>
```

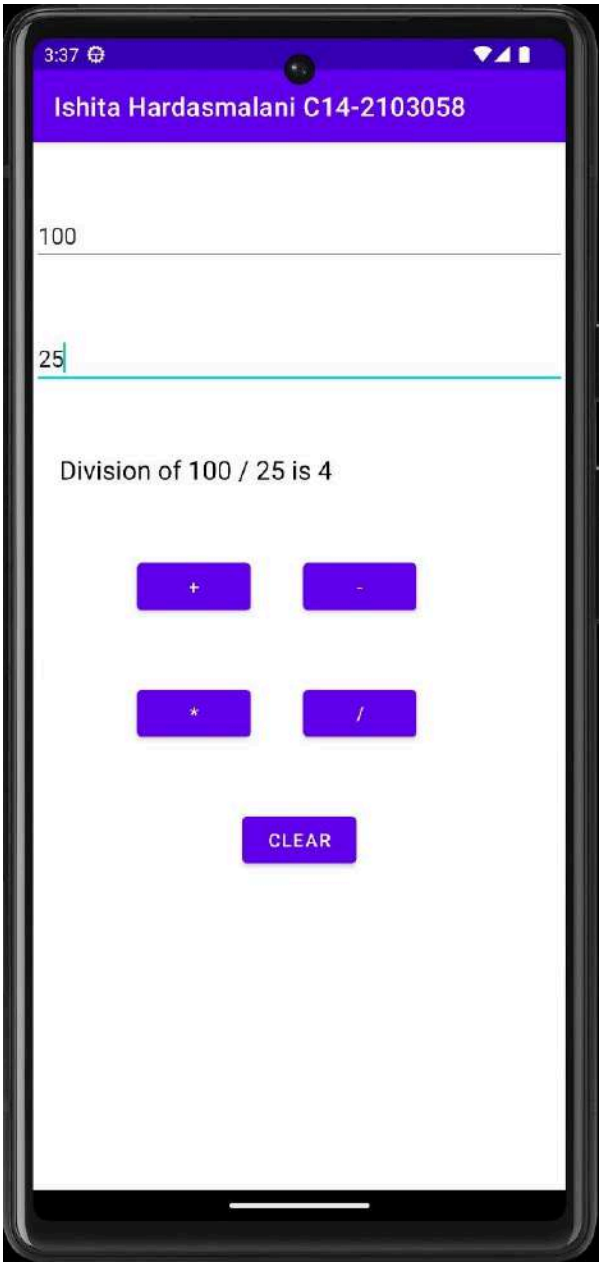
OUTPUT:


3:35
Ishita Hardasmalani C14-2103058
First Number
Second Number
+
-
*
/
CLEAR


3:36
Ishita Hardasmalani C14-2103058
100
25
Addition of 100 + 25 is 125
+
-
*
/
CLEAR


3:37
Ishita Hardasmalani C14-2103058
100
25
Subtraction of 100 - 25 is 75
+
-
*
/
CLEAR


3:37
Ishita Hardasmalani C14-2103058
100
25
Multiplication of 100 * 25 is 2500
+
-
*
/
CLEAR


3:37
Ishita Hardasmalani C14-2103058
100
25
Division of 100 / 25 is 4
+
-
*
/
CLEAR

Experiment No. 8

Aim: write a program to demonstrate cellular frequency reuse.

Theory:

cellular frequency reuse is cellular network design that allows for efficient use of limited frequency resources. the basic idea is to divide the available frequency spectrum into smaller chunks called channel & then reuse these channel accross different cells within the network to maximize the overall capacity & coverage.

using android devices to implement cellular frequency reuse involves several steps:

1. Frequency planning: the first step is to plan the allocation of frequency channels to different cells in the network. this typically involves dividing the coverage area into smaller cells & assigning non-overlapping frequency channel to each cell. the goal is to minimize interference between adjacent cells while maximizing the overall capacity of network.
2. Android device configuration: Android devices such as smartphones & tablets, are equipped with cellular radios

that operate within specific frequency bands allocated by regulatory authorities. These devices need to be configured to use the appropriate frequency channel based on the frequency planning done in step 1. This can be done either manually by user or automatically by the devices software.

3. Interference Management: One of the key challenges in cellular frequency is managing interference between adjacent cells that share the same frequency channels. Android device can help mitigate interference by employing advanced signal processing techniques such as interference cancellation & adaptive modulation.

4. Dynamic frequency allocation: In a dynamic cellular network environment, the allocation of frequency channel to cells may need to be adjusted in real-time based on changing traffic patterns & traffic conditions. Android devices can play a role in this by providing feedback to the network infrastructure about signal strength, quality & other relevant metrics.

5. Monitoring & optimization: Finally android devices can be used to monitor the performance of the cellular network & identify area where frequency reuse can be further optimized. This may involve

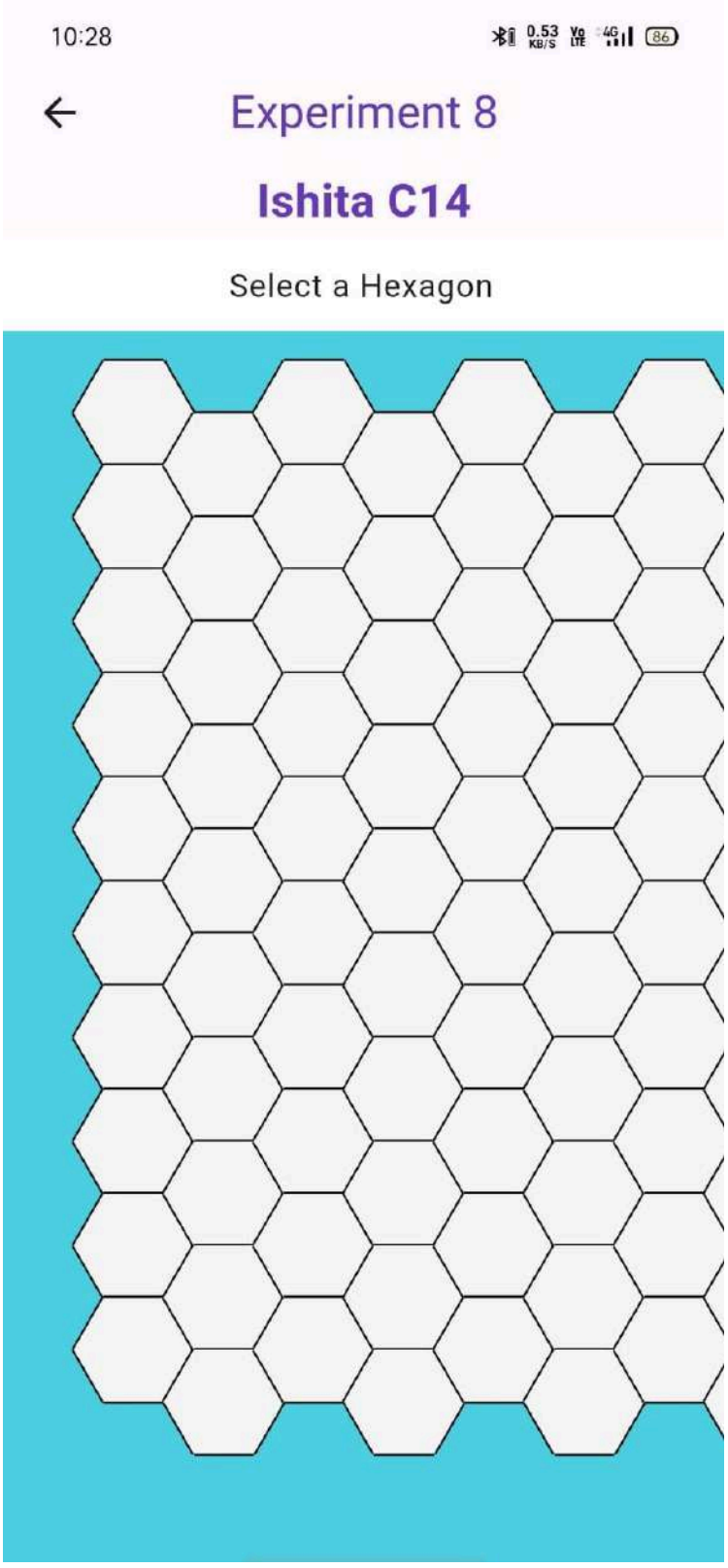
conducting site surveys, analyzing network traffic pattern & perform drive tests to access signal strength & quality.

MCC EXPERIMENT 8

```
import java.util.HashMap;
import java.util.Map;

public class CellularFrequencyReuseDemo {

    // Represents a single cell in the network
    static class Cell {
        int cellId;
        int frequencyChannel;

        public Cell(int cellId, int frequencyChannel) {
            this.cellId = cellId;
            this.frequencyChannel = frequencyChannel;
        }
    }

    public static void main(String[] args) {
        // Create a network with multiple cells
        Cell[] cells = {
            new Cell(1, 1),
            new Cell(2, 2),
            new Cell(3, 1),
            new Cell(4, 2),
            new Cell(5, 3),
            new Cell(6, 1)
        };

        // Map to store frequency channels and their corresponding cells
        Map<Integer, Cell> frequencyChannels = new HashMap<>();
```

```
        // Demonstrate frequency reuse
        for (Cell cell : cells) {
            // Check if the frequency channel is already allocated
            if (frequencyChannels.containsKey(cell.frequencyChannel)) {
                // If the channel is already allocated, print the reuse information
                System.out.println("Cell " + cell.cellId + " reuses frequency channel " +
cell.frequencyChannel +
                                   " (already allocated to Cell " +
frequencyChannels.get(cell.frequencyChannel).cellId + ")");
            } else {
                // If the channel is not allocated, allocate it to the current cell
                frequencyChannels.put(cell.frequencyChannel, cell);
                System.out.println("Cell " + cell.cellId + " uses frequency channel " + cell.frequencyChannel);
            }
        }
    }
}
```

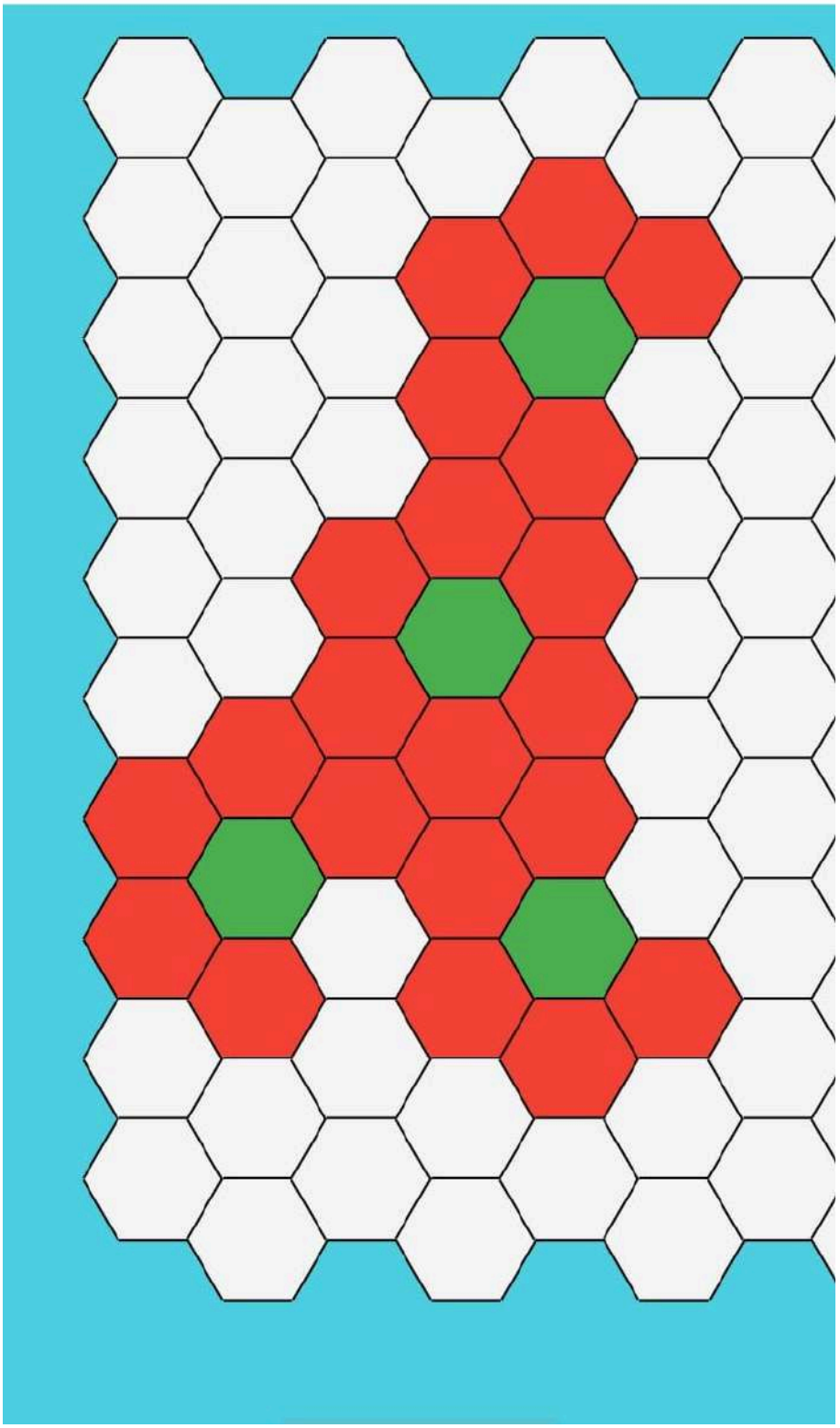
Output:-



Experiment 8

Ishita C14

Hexagon selected: 7,5



Experiment No. 9

Aim: Write a program to explain concept of DSSS.

Theory:

Direct sequency spread spectrum (DSSS) is a modulation technique used in wireless communication to spread the signal over a wider bandwidth than the original information bandwidth. This is achieved by multiplying the data signal with a high-frequency pseudo-random bit sequence known as a spreading code. The spreading code has a much higher data rate than the original data signal, effectively spreading the signal across a larger frequency range.

To implement DSSS steps are:

1. Spreading code generation: The first step in implementing DSSS is to generate a spreading code. This code is a pseudo-random sequence of bits with a much higher data rate than the original data signal.
2. Modulation: The spreading code is then used to modulate the original data signal. This involves multiplying each bit of the spreading code

3. Transmission: The modulated signal is transmitted over the communication channel. Because the signal is spread over a wider bandwidth, it is more resilient to interface & noise.

4. Reception: At the receiver, the received signal is despread using the same spreading code that was used for modulation. This involves multiplying the received signal by the spreading code.

5. Error correction: DSSS system often incorporate error correction techniques such as forward error correction to improve the reliability of data transmission.

To implement DSSS in a wireless communication we need to design & implement these components.

Spreading code generator
Modulation circuitry
Transmitter
Receiver
error correction.

Ishita Hardasmalani C14 58

Experiment 9

Code:

```
import java.lang.*;
import java.io.*;
import java.util.*;

class dsss {

	public static void main(String args[]) {
		Scanner scanner = new Scanner(System.in);
		System.out.print("Enter input string: "); String
		inputString = scanner.next();
		String barcaCode = "10110111000";

		// Encryption
		String eOutput = "";
		for(int i=0; i<inputString.length(); i++) {
			String a = getString(inputString.charAt(i));
			eOutput = eOutput + getEXOR(a, barcaCode);
		}
		System.out.println("\nEncrypted message: "+eOutput+"\n");


		// Decryption
		ArrayList<String> enStrings = new ArrayList();
		for(int i=0; i<eOutput.length(); i=i+11) {
			enStrings.add(eOutput.substring(i,i+11));
		}
		String dOutput = "";
		for(int i=0; i<enStrings.size(); i++) {
			String a = getEXOR(enStrings.get(i),barcaCode);
			if(getNoOfOnes(a)>7) {
				dOutput = dOutput + "1";
			} else if(getNoOfOnes(a)<3) {
				dOutput = dOutput + "0";
			}
		}
		System.out.println("Decrypted message: "+dOutput+"\n\n"); }
	//  Method for getting 11111111111 or 00000000000
	public static String getString(char a) {
		if(a=='1') {
			return "11111111111";
		} else {
			return "00000000000"; }
	}
	// Method for performing ex-or
```

```
	public static String getEXOR(String x, String y) {

		String z = "";
		for(int i = 0; i<x.length(); i++) {
			if((x.charAt(i)=='1' && y.charAt(i)=='1') || (x.charAt(i)=='0' &&
y.charAt(i)=='0')) {
				z = z + "0";
			} else if ((x.charAt(i)=='0' && y.charAt(i)=='1') || (x.charAt(i)=='1' &&
y.charAt(i)=='0')) {
				z = z + "1";
			}
		}
		return z;
	}
    /* Method for getting number of one's in string */
	public static int getNoOfOnes(String a) {
		int count = 0;
		   for(int i=0; i<a.length();i++) {
			if(a.charAt(i) == '1') {
				count = count + 1;
			}
		}
	return count;
	}
}
```

Output 01:

```
PS D:\Engineering\sem 6\Ishita_Sem_6\MCC\Experiments\DSSS> java dsss
Enter input string: 100101

Encrypted message:
0100100011110110111000101101110000100100011110110111000010010001
1

Decrypted message: 100101
```

Output 02:

```
PS D:\Engineering\sem 6\Ishita_Sem_6\MCC\Experiments\DSSS> java dsss
Enter input string: 1110101

Encrypted message:
0100100011101001000111010010001111011011100001001000111101101110000100100011
1

Decrypted message: 1110101
```

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

A+

EXPERIMENT NO. 10.

Aim: write a program to implement A3/A5/A8 GSM security algorithms

Theory:

The A3, A5 & A8 algorithm are security algorithms used in the global system for mobile communication to provide confidentiality integrity & authentication for the user & signaling messages. This algorithm are designed to protect the privacy of user communication & prevent unauthorised access to the network.

- A3 Algorithm (GSM confidentiality Algorithm).
- A3 is used for encrypting user data transmitted over air interface between the mobile device & the base station (BTS).
- The algorithm takes as input a 64-bit cipher key (KC) the data to be encrypted. It then generates a sequence of keystream bits using a linear feedback shift reg. algorithm.
- The generated keystream is XORed with plaintext to produce the ciphertext, which is transmitted over the air interface.
- After the receiver, the same algorithm is applied to the same key to decrypt the received ciphertext & recover original plaintext data.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

2. A15 Algorithm (asm encryption algorithm).
A15 is most widely used encryption algorithm in GSM network. It is used to encrypt the signal messages exchanged between the mobile device & the base station, including authentication & key agreement messages.
A15 operates in two modes: A151 & A1512. A151 is the stronger version while A512 is weaker & less secure.
The algorithm takes as input: a 64 bit cipher key (KC) & the data to be encrypted. It then performs a series of logical & arithmetic operation on the data to process cipher text.

3. A18 algorithm (GSM Authentication Algorithm).
A18 is used for authentication mobile device when they attempt to access the GSM network.
It is based on a challenge response received from mobile device to authenticate it. key (Ki) & the A18 algorithm to generate response.
The network independently calculates SRES using the same key & compares it with response received from the mobile device to authenticate it.

Ishita Hardasmalani C14 58

MCC EXPERIMENT 10

```
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;

public class AESExample {

  public static void main(String[] args) {
    try {
      String originalText = "Hello, world!";
      String secretKey = "ThisIsASecretKey";

      String encryptedText = encrypt(originalText, secretKey);
      System.out.println("Encrypted text: " + encryptedText);

      String decryptedText = decrypt(encryptedText, secretKey);
      System.out.println("Decrypted text: " + decryptedText);
    } catch (Exception e) {
      e.printStackTrace();
    }
  }

  public static String encrypt(String input, String key) throws Exception {
    Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
    SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), "AES");
    cipher.init(Cipher.ENCRYPT_MODE, secretKey);
    byte[] encryptedBytes = cipher.doFinal(input.getBytes());
    return Base64.getEncoder().encodeToString(encryptedBytes);
  }
```

```
    public static String decrypt(String input, String key) throws Exception {
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        SecretKeySpec secretKey = new SecretKeySpec(key.getBytes(), "AES");
        cipher.init(Cipher.DECRYPT_MODE, secretKey);
        byte[] decryptedBytes = cipher.doFinal(Base64.getDecoder().decode(input));
        return new String(decryptedBytes);
    }
}
```

Output:-

Encrypted text: kSLBfmxk26e5UpYyf7Qzdg==

Decrypted text: Hello, world!

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

WRITTEN ASSIGNMENT -1

Q1. Write short notes on antenna

Ans: Antennas, integral to wireless communication transmit & receive electromagnetic waves. It plays a crucial role in wireless communication.

Functions of antenna:

1. Transmitting antennas convert electrical signals into electromagnetic waves for transmission into the air.
2. Receiving antennas capture incoming electromagnetic waves & convert them into electrical signals for further processing.

Types of antenna:

1. Dipole antenna: Simplest form consisting of two conductive elements.
2. Isotropic antenna: It is a theoretical antenna, it radiates its power uniformly in all directions. It has a perfect 360° spherical radiation. It is used to compare the power-level of a given antenna to the isotrophic antenna.
3. Directional Antenna: It is also called as Beam Antenna. It is an antenna which radiates or receives greater power in specific directions. This allows increased performance & reduced interference from unwanted sources.

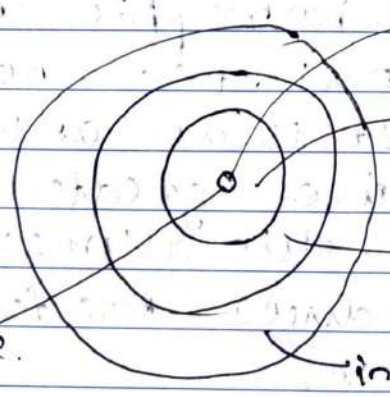
characteristics:

1. Gain: Measures the ability of an antenna to direct or concentrate the transmitted or received signal.
2. Directionality: Antennas can be omnidirectional or directional.
3. Polarization: Refers to the orientation of the electric field in the electromagnetic wave.

Q2. What are various issues in signal propagation.

Ans.a Wired networks confine signals to wires, while wireless networks propagate signals through unguided media like radio waves.

b. Depending on the distance from the sender, the transmitted signal can fall in the following ranges.



1. Transmission Range:
→ senders can transmit the signals very easily.
→ Minimum or no power required.
→ Minimum or no error rate & background noise.

2. Detection range:
→ More power is required
→ High error rate.

3. Interference Range:
→ High chances of signals getting interrupted.
→ signal may or may not reach receiver.

c. Free space loss: As the signal propagate through space, their power diminishes with increasing distance, following inverse square law.

d. Also, factors such as wavelength & antenna gain, play crucial roles in determining the strength of received signal.

e. Reflection: signals encountering large surfaces undergo reflection, where a portion of the signal is redirected away from the receiver leading to signal loss & potential interference.

f. Refraction: Changes in density cause signal bending, altering the path,

g. Scattering: Signals interacting with objects smaller than their wavelength scatter in various directions, causing signal dispersion & potentially reducing signal strength.

Q3. Explain different application of mobile computing.

Ans. 1) Mobile banking: Accessing bank services through mobile apps for transactions, balance enquiries & payments.

2) Navigation & maps: GPS enabled mobile devices for real time navigation, mapping, & location based services.

3) E-commerce: Mobile platforms for online shopping enabling users to browse, purchase & track deliveries.

4) Social media: Mobile apps for social networking facilitating communication, & current location.

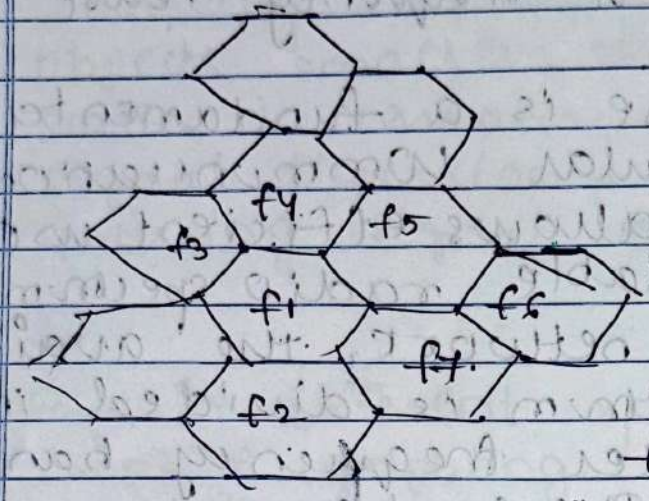
5) Emergency services: Mobile apps in emergency situations with features like SOS calls, location sharing & alerts

Q4. Explain concept of frequency reuse with clustering.

Ans Frequency reuse is a fundamental concept in cellular communication systems that allows efficient use of the available radio spectrum. In a cellular network, the available frequency spectrum is divided into multiple smaller frequency bands or channels. These channels are then reused across different cells within the network, following a specific pattern to minimize interference & maximize spectral efficiency. To avoid interference in cellular system, each cell uses a different set of frequencies as compared to its immediate neighbors. A set of several cells are further grouped into clusters. Cells within the same cluster do not use the same frequency sets. Consider a cellular system has 'S' full duplex channels for use. Assume that the 'S' channels are divided into 'N' number of cells & each cell is allocated a group of 'K' channels ($K < S$).

∴ Total number of channels per cell is

$K = S/N$.



4 cell cluster.

The N cells which collectively use the complete set of available frequencies is called cluster. The factor N is called the cluster size. It is typically 4, 7 or 12. Frequency reuse factor of a cellular system is given by reciprocal of the cluster size i.e. $1/N$.

If the cluster size N is reduced while the cell size remains constant; more cluster are required to cover that particular area & hence more capacity is achieved.

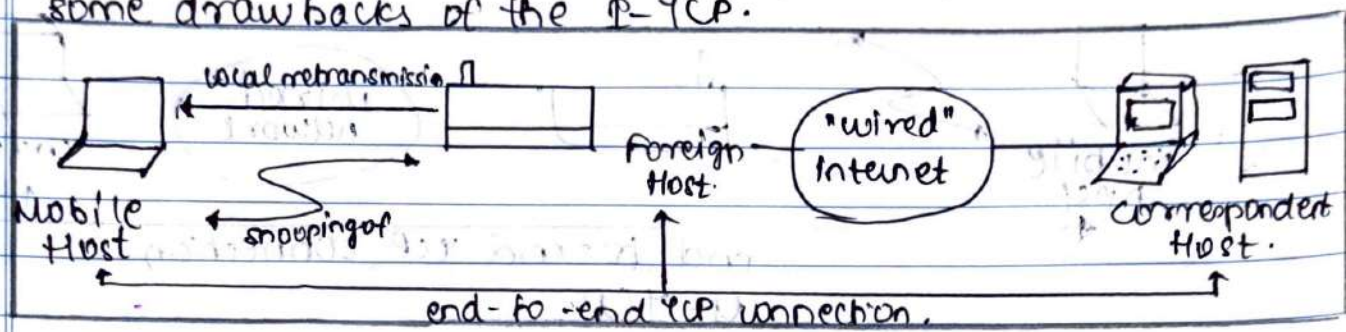
THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Written Assignment - 2

(A)

Explain in detail with merits & demerits.

a) Snooping TCP : It works completely transparently & leaves the TCP end-to-end connection intact. It overcomes some drawbacks of the I-TCP.



SNOOPING-TCP.

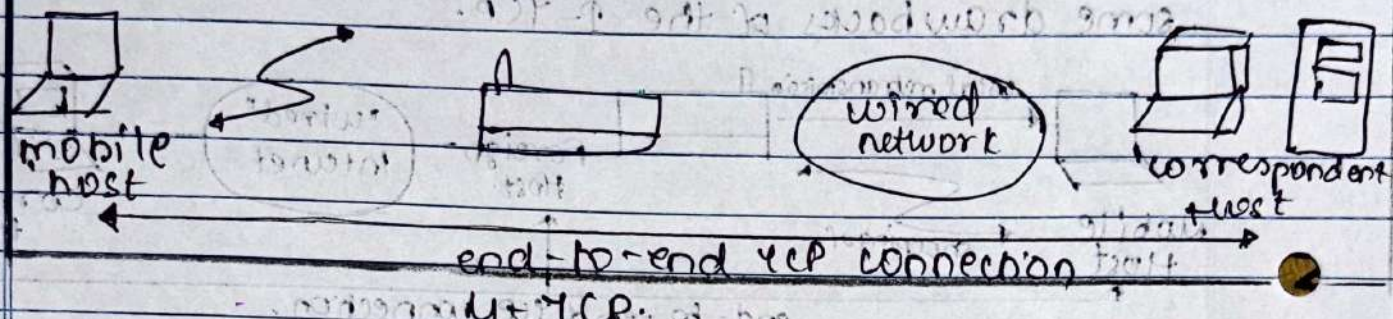
Snooping TCP works as follows:

1. Correspondent host sends a packet to mobile host: It's sent via wired TCP connection. The accesspoint buffers the packet sent by correspondent host. Accesspoint also snoops on the packet in both directions to reorganize acknowledgements.
2. mobile host transmits a packet to a correspondent host: when a mobile host sends a packet to correspondent host, foreign agent keeps track of the sequence numbers of these packets.

Advantages: → end-to-end semantics are preserved
→ correspondent host need not to be changed enhancements are done in foreign agent.

Disadvantages: → effectiveness of S-TCP completely depends on the quality of wireless link.
→ If user applies end-to-end encryption, S-TCP fails, because TCP header would be encrypted & hence snooping on the sequence number is meaningless

b) **Mobile TCP**: The occurrence of lengthy and/or frequent disconnection is the major problem in wireless networks. M-TCP deals with the lengthly and/or frequency disconnections.



M-TCP.

M-TCP aims:

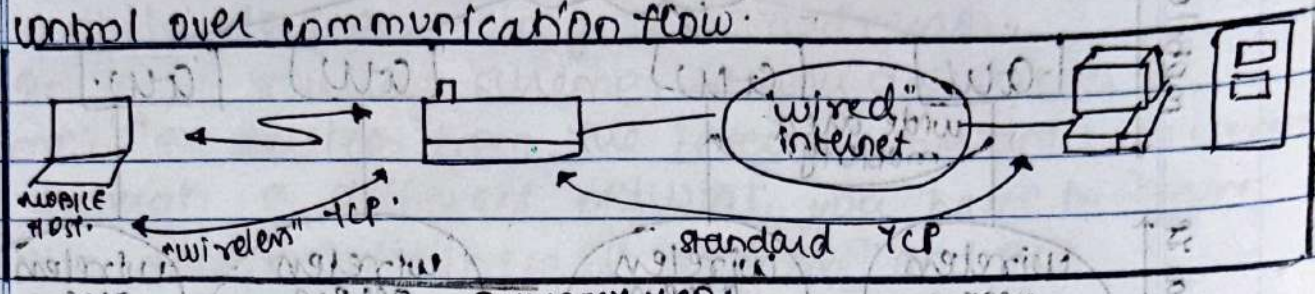
- To improve overall throughput
- To lower the delay
- To maintain end-to-end semantics of TCP
- To provide more efficient handover

Challenges in mobile networks like 3G, 4G & 5G.
Mobile TCP variants like Hybla, Vegas & Bullions aim to enhance adaptability to fluctuating mobile conditions.

Advantages: → It maintains TCP end-to-end semantics
→ Lost packets are automatically retransmitted to the new BH.

Disadvantages: → Packet loss on wireless link due to errors is propagated to the sender because BH does not act as proxy.
→ M-TCP always assumes low bit error rates, which are not always valid assumption.

c) I-TCP (Indirect TCP) : It is a variant optimized for satellite communication networks, addressing challenges introduced by satellite latency. It splits the TCP connection into forward & reverse connections for better control over communication flow.



I-TCP provides more reliable communication over satellite networks by optimizing congestion control.

Advantages : → I-TCP does not require any change in TCP protocol. → In "I-TCP optimized TCP could use precise time-outs to guarantee retransmission more faster.

→ Different transport layer protocol can be used between mobile host & foreign host.

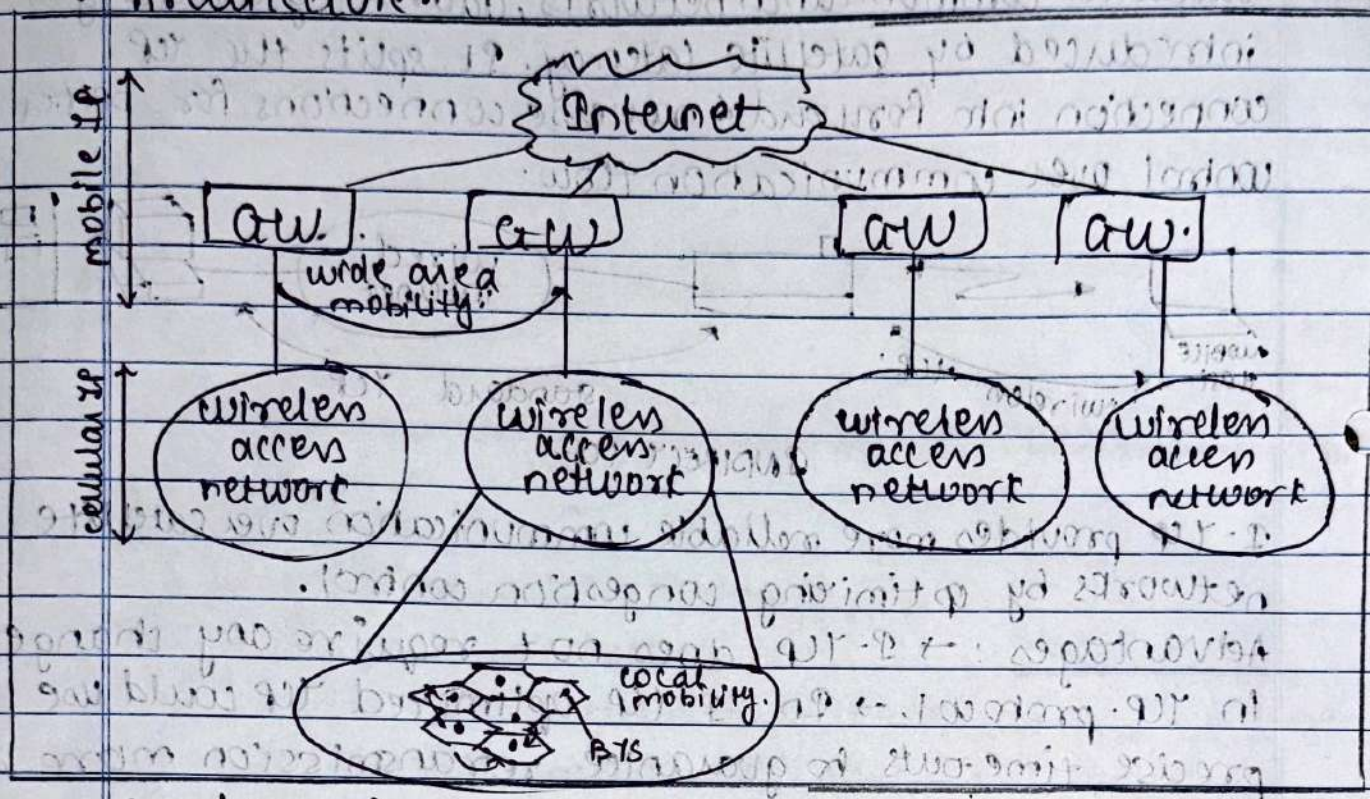
Disadvantages : Increased handover latency is problematic since foreign agent not necessarily forwards packets sent by correspondent host to mobile host. The foreign agent is required to be trusted entity because TCP connection end at this point.

Accurate RTT estimation considers satellite propagation delays, preventing unnecessary slowdowns.

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Q2: Explain cellular (mobile) IP in detail.

• Architecture:



consists of 3 major components.

- cellular IP gateway (GW)
- cellular IP node or the baise station (BS)
- cellular IP mobile host (MH).

An important component of cellular IP network is the base station.

Routing in cellular IP manages mobile device management between cells using routing protocols, optimizing data delivery. It dynamically reroutes traffic as device moves.

Paging in IP alerts mobile devices of incoming data or calls by broadcasting paging messages.

Handover in cellular transitions of mobile devices connection from one base station to another as it moves

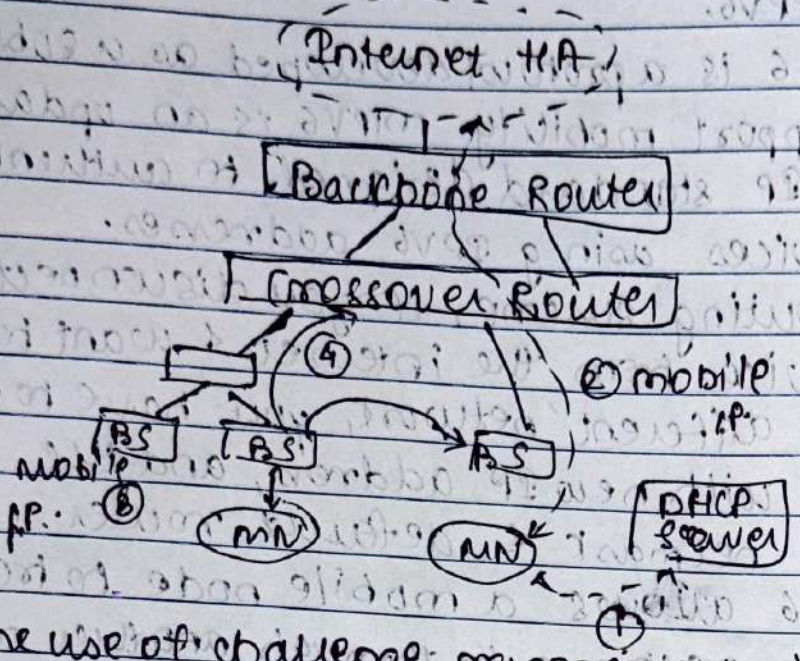
Advantages: → Provides easy global migration
→ cheap passive connectivity.

Q3. Explain MIPV6.

- Mobile IPV6 is a protocol developed as a subset of IPV6 to support mobility. MIPV6 is an update of the mobile IP standard designed to authenticate mobile devices using IPV6 addresses.
- In this routing scheme, if you disconnect a mobile device from the internet & want to reconnect through a different network, you have to figure the device with new IP address and the appropriate netmask & default router.
- Mobile IPV6 allows a mobile node to transparently maintain connections through another network.
- When connecting through a foreign network, a mobile device sends its location information to a home agent; which intercepts packets intended for the device & tunnels them to the current location
- Each device is identified by its home address although it may be connecting to through another network.

Q4. Write short notes on : HAWAII.

HAWAII (Handoff-Aware Wireless Access Internet Infrastructure) tries to keep micromobility support as transparent as possible for both home agents & mobile nodes. It concrete goals are performance & reliability improvements & support for quality of service mechanisms. On entering an HAWAII domain, a mobile node obtains a co-located CoA & registers with the HA.



The use of challenge response extensions for authenticating a mobile node is mandatory. HAWAII claims to be mostly transparent to mobile nodes.

Advantages:

→ Security: challenge-response extensions are mandatory.

→ Transparency: HAWAII is mostly transparent to MN

Disadvantages:

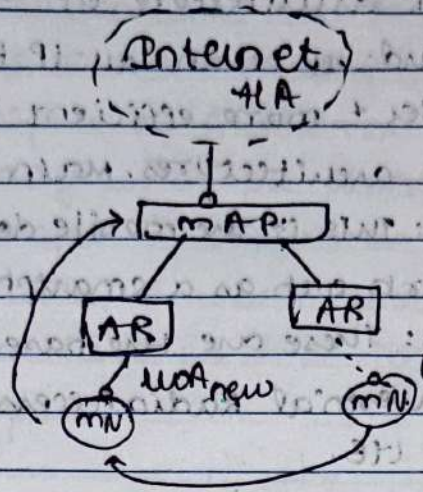
→ Security: there is no provision regarding the setup of IPSec tunnels.

→ Implementation: NO private address support is possible.

Q5. Explain HMIPV6 in detail.

Hierarchical mobile IPV6 (HMIPV6) provides micro-mobility support by installing a mobility anchor point (MAP). It is an entity which is responsible for a certain domain & acts as a local HA within this domain for visiting MNs. All the MAP receives all packets on behalf of the MN

Architecture:



Advantages:

→ Security: MNs can have location privacy because CoAs can be hidden.

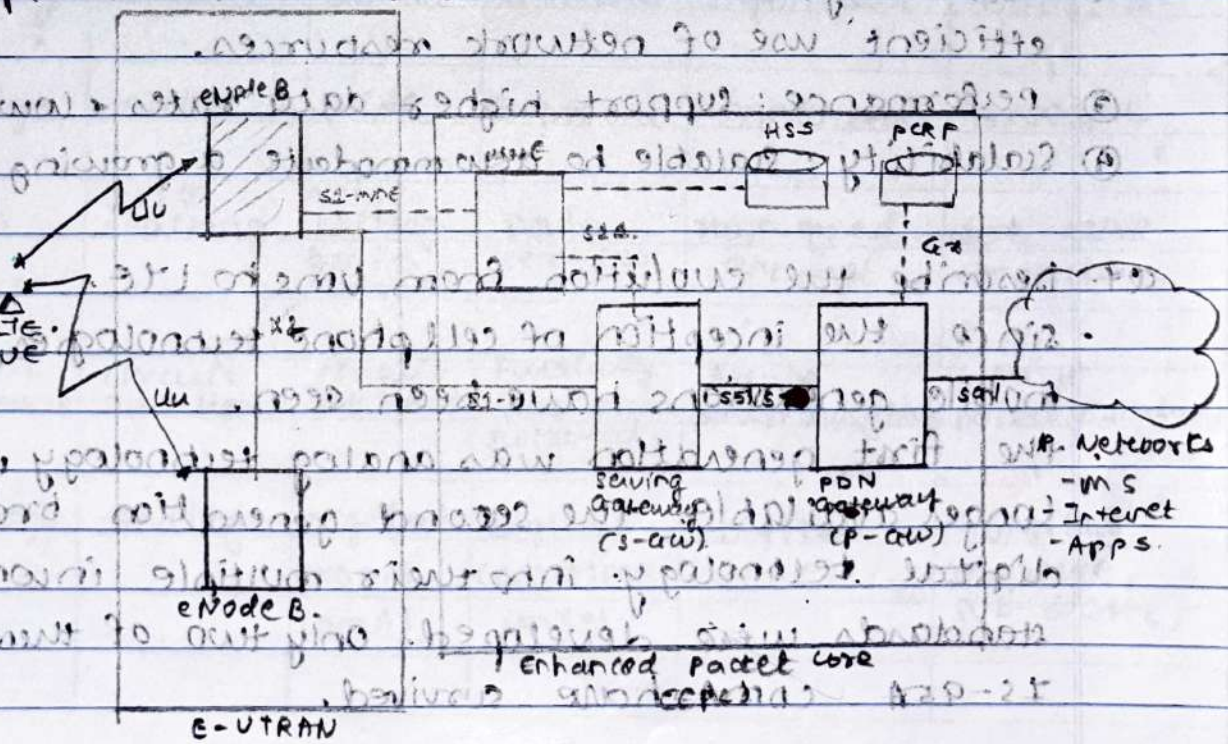
→ Efficiency: Direct routing b/w CNs having same link.

Disadvantages:

→ Transparency: Additional infrastructure component.

→ Security: Additional security functions for AAAs.

Q6. Explain SAE architecture.



- It is the core network architecture of 3GPP's LTE wireless communication standard. It is an all-IP based, flat architecture designed to be simpler & more efficient than previous generations of cellular network architectures. Main components of SAE archi.
- User Equipment (UE): This is the mobile device that you use to connect to the network such as a smartphone, or tablet.
- Evolved NodeB (eNB): These are the base stations that make up the Evolved UMTS Terrestrial Radio Access Network, the radio access network for LTE.
- Evolved Packet Core (EPC): This is the core network of the SAE.
- Mobility management entity (MME): This is responsible for managing the connection between the UE & the network.
- Serving Gateway (S-GW): Responsible for routing & forwarding user data packets between the UE & PDN Gateway.
- Packet Data Network Gateway: point of connection between the UE & external networks such as Internet

Advantages: ① Simplicity - less expensive to deploy & operate.
② Efficiency: The all-IP architecture of SAE allows for more efficient use of network resources.
③ Performance: Support higher data rates & lower latency.
④ Scalability: Scalable to accommodate a growing no. of users.

Q4. Describe the evolution from UMTS to LTE.

- Since the inception of cellphone technologies, many mobile generations have been seen.
- The first generation was analog technology which is no longer available. The second generation brought digital technology into their multiple incompatible 2G standards were developed. Only two of them, GSM & IS-95A CDMA have survived.

the third generation (3G) standard came into market. Again, multiple standards were developed, mainly, WCDMA by the 3GPP & CDMA 2000 by Qualcomm. Both have survived & are still used today. The 3G standards were continually updated to what is known as 3.5G. WCDMA was upgraded to HSPA, & CDMA 2000 was expanded with 1xRTT EV-DO releases A & B. The 3GPP developed the widely used UMTS.

Q8. Compare mobile generations (1G, 2G, 3G, 4G, 5G).

	1G	2G	3G	4G	5G
Data Speed	very low (2.4 kbps)	upto 64 kbps	several mbps	10-100 mbps	multi Gbps.
Latency	high	high	10-1000 ms	30-50 ms	1 ms or lower
Capacity	limited	limited	Improved	Significant	Further boosted.
Coverage	wide	wide	Improved	Enhanced	Broader
Use case	voice calling	voice calling, basic data	Data services	High speed internet	IoT, AR/VR connection.
Network Architecture	Circuit switched	Circuit switched	Partially packet switched	All-IP packet switched	All-IP packet switched.
Technology	Analog	Digital (CDMA, GSM)	Digital (CDMA 2000, UMTS)	Digital (LTE)	Digital (mmWave, Sub-6 GHz)

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Q9. What are self-organizing networks.

Self organizing networks are telecommunications systems that automate the configuration, optimization & management of wireless networks. They utilize intelligent algorithms & automation to adapt to changing network conditions & enhance performance. SON's handle tasks such as parameter optimization, interference management, load balancing & fault detection. They improve service quality, reduce operational costs & minimize human errors. Essential for managing the complexity of modern wireless networks, including 4G LTE, 5G & beyond. Reduced operational costs, Improved network performance, Faster network deployment, Increased network agility.

Q10. Explain VoLTE in detail.

VoLTE stands for Voice over Long-Term Evolution. It is a technology that enables voice calls to be carried over the LTE data network instead of the traditional circuit switched networks used in 2G & 3G. Key aspects of VoLTE:

1. IP-Based voice transmission: VoLTE utilized packet switched technology to transmit voice as data packets.
2. Enhanced voice quality: Delivers high-definition voice quality, providing clearer & more natural sounding audio.
3. Simultaneous voice & Data: Allows users to make voice calls
4. Efficient Spectrum utilization: Designed to use network resources more efficiently.
5. Global Adoption: VoLTE adoption has been growing globally as mobile operators transition to more advance network technologies.