Laboratory Manual for

Mobile Application Development (3170726)

B.E. Semester 7
Computer Engineering

GEC Dahod

Directorate of Technical Education, Gujarat

Name	Enrollment Number
Jainab khatri	200180107002

Government Engineering College, Dahod

Certificate

This is to certify that Mr./Ms. _____ Enrollment No. _____of B.E. Semester _____ Computer Engineering of this Institute (GTU Code: 018) has satisfactorily completed the Practical / Tutorial work for the subject _____for the academic year 2022-23.

Place: _____

Date: _____

Name and Sign of Faculty member

Head of the Department

Preface

Welcome to the exciting world of mobile application development! In an era where smartphones and tablets have become an integral part of our lives, the ability to create innovative and user-friendly mobile applications is a valuable skill that transcends industries and empowers individuals to shape the digital landscape.

This course is designed to take you on a journey through the fundamentals and intricacies of mobile application development. Whether you are a seasoned coder or a beginner with a passion for technology, this subject will equip you with the knowledge and skills needed to design, build, and deploy mobile apps that stand out in today's competitive app ecosystem.

By using this lab manual students can go through the relevant theory and procedure in advance before the actual performance which creates an interest and students can have basic idea prior to performance. This in turn enhances pre-determined outcomes amongst students. Each experiment in this manual begins with competency, relevant skills, course outcomes as well as practical outcomes (objectives). The students will also achieve safety and necessary precautions to be taken while performing practical.

This manual also provides guidelines to faculty members to facilitate student centric lab activities through each experiment by arranging and managing necessary resources in order that the students follow the procedures with required safety and necessary precautions to achieve the outcomes. It also gives an idea that how students will be assessed by providing rubrics.

Practical – Course Outcome matrix

Course Outcomes (COs):	
CO1	Understand Android architecture, activities and their life cycle.
CO2	Apply the knowledge to design user interface using Android UI And Component
CO3	Manage system database, remote database operations using web services and Firebase
CO4	Apply knowledge of map, location services, Graphics, android system and background services
CO5	Publish and distribute Android Application

Sr. No.	Objective(s) of Experiment	CO 1	CO 2	CO 3	CO 4	CO 5
1.	Design Login activity and implement control events: Use EditText, Checkbox and Buttons. And implement above events using following layouts: 1. Linear Layout 2. Relative Layout 3. Table Layout		√			
2.	Create Activities & implement following 1. Implicit intent 2. Explicit Intent 3. Start Activity for Result	√	√			
3.	Practical: Use an Options Menu		√			
4.	Create a Recycler View and list the details of student using following fields: 1. Name 2. Address 3. Photo (Image) 4. Delete (Button Operation)	√	√			
5.	Practical: Theme, Custom Styles, Drawable			√		
6.	Practical: Save user data in a database			√		
7.	Practical: Get and Save User Preferences			√	√	
8	Practical: make a use of android system					√
9	Using location service get the current location and display in Text View			√	√	
10	Practical: Display the use of animations				√	

Industry Relevant Skills

The following industry relevant competency are expected to be developed in the student by undertaking the practical work of this laboratory.

1. Investigation and analysis skills: Develop the ability to investigate and analyze various digital devices and systems, including computers, mobile devices, and networks. Learn how to extract and analyze data from these devices and systems to identify evidence of cybercrime.
2. Evidence handling and preservation skills: How to handle and preserve digital evidence in a way that is admissible in court. This includes learning about chain of custody, evidence storage, and documentation.
3. Technical skills: Technical skills related to computer and network security, including knowledge of operating systems, file systems, and network protocols. Students may also learn about encryption, steganography, and other techniques used to hide information.
4. Legal and regulatory knowledge: Relevant laws and regulations related to cybercrime, such as the IT Act 2000. Students will learn about legal procedures, courtroom procedures, and other aspects of the legal system.
5. Communication and reporting skills: Students will learn how to communicate complex technical information to non-technical stakeholders, such as lawyers, judges, and juries. They will also learn how to write clear and concise reports that summarize their findings and conclusions.
6. Critical thinking and problem-solving skills: Complex problem-solving scenarios that require students to think critically and apply their knowledge and skills to real-world situations.

Guidelines for Faculty members

1. Teacher should provide the guideline with demonstration of practical to the students with all features.
2. Teacher shall explain basic concepts/theory related to the experiment to the students before starting of each practical
3. Involve all the students in performance of each experiment.
4. Teacher is expected to share the skills and competencies to be developed in the students and ensure that the respective skills and competencies are developed in the students after the completion of the experimentation.
5. Teachers should give opportunity to students for hands-on experience after the demonstration.
6. Teacher may provide additional knowledge and skills to the students even though not covered in the manual but are expected from the students by concerned

industry.

7. Give practical assignment and assess the performance of students based on task assigned to check whether it is as per the instructions or not.
8. Teacher is expected to refer complete curriculum of the course and follow the guidelines for implementation.

Instructions for Students

1. Students are expected to carefully listen to all the theory classes delivered by the faculty members and understand the COs, content of the course, teaching and examination scheme, skill set to be developed etc.
2. Students shall organize the work in the group and make record of all observations.
3. Students shall develop maintenance skill as expected by industries.
4. Student shall attempt to develop related hand-on skills and build confidence.
5. Student shall develop the habits of evolving more ideas, innovations, skills etc. apart from those included in scope of manual.
6. Student shall refer technical magazines and data books, follow real cyber forensic cases.
7. Student should develop a habit of submitting the experimentation work as per the schedule and s/he should be well prepared for the same.

Common Safety Instructions

Students are expected to carefully perform each experiment without damaging the lab computer systems. All the experiments are for learning purpose only and never perform anywhere else without proper authorization.

Rubrics used for Evaluation 10 marks for each practical assignment.

Knowledge of subject (2)		**Programming Skill**		**Team work (2)**		**Communication Skill (2)**		**Ethics(2)**	
Good (2)	Average (1)	Good (2)	Average (1)	Good (2)	Satisfactory (1)	Good (2)	Satisfactory (1)	Good (2)	Average (1)

Index
(Progressive Assessment Sheet)

Sr. No.	Objective(s) of Experiment	Page No.	Date of perf orm ance	Date of sub miss ion	Asses sment Mar ks	Sign. of Teac her with date	Rem arks
0	Write the Following 1. Vision & Mission of DTE, GEC Dahod and Computer Department 2. Program Outcome of Computer Engineering 3. PSOs and PEOs of Computer Engineering Department 4. Course outcomes of Web Programming						
1.	Design Login activity and implement control events: Use EditText, Checkbox and Buttons. And implement above events using following layouts: 1. Linear Layout 2. Relative Layout 3. Table Layout						
2.	Create Activities & implement following 1. Implicit intent 2. Explicit Intent 3. Start Activity for Result						
3.	Practical: Use an Options Menu						
4.	Create a Recycler View and list the details of student using following fields: 1. Name 2. Address 3. Photo (Image) 4. Delete (Button Operation)						
5.	Practical: Theme, Custom Styles, Drawable						
6.	Practical: Save user data in a database						
7.	Practical: Get and Save User Preferences						
8	Practical: make a use of android system						
9	Using location service get the current location and display in Text View						
10	Practical: Display the use of animations						
	Total						

Vision and Mission of Computer Department:

Vision:

- ✓ To be a distinguished department that provides conducive environment for learning, innovation and research while nurturing an inclusive and all-round development of its stakeholders.

Mission:

- ✓ To prepare computer engineers capable of implementing professionally valued solutions.
- ✓ To imbibe aspect of life long learning, adapting technological change and addressing needs of society by keeping abreast of advancements in engineering and technology.
- ✓ To adhere to core ethical values, social values and sensitivity towards environment.

Practical-1

AIM: Design Login activity and implement control events: Use EditText, Checkbox and Buttons. And implement above events using following layouts:

1. Linear Layout 2. Relative Layout 3. Table Layout

Code: XML FILE, JAVA FILE

→Activity_.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:background="#FFFFFF">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:orientation="vertical">
        <EditText
            android:id="@+id/username"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="UserName" />
        <EditText
            android:id="@+id/password"
            android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:hint="password"
        android:inputType="textPassword" />
    <Button
        android:id="@+id/login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#0F3CAB"
        android:text="Login"
        android:textColor="#FFFFFF" />
</LinearLayout>
<TextView
    android:id="@+id/textView8"
    android:layout_width="203dp"
    android:layout_height="29dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_marginStart="10dp"
    android:layout_marginTop="52dp"
    android:layout_marginEnd="188dp"
    android:layout_marginBottom="10dp"
    android:text="Name: Jainab Khatri"
    android:textSize="16sp" />
<TextView
    android:id="@+id/textView9"
    android:layout_width="234dp"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
```

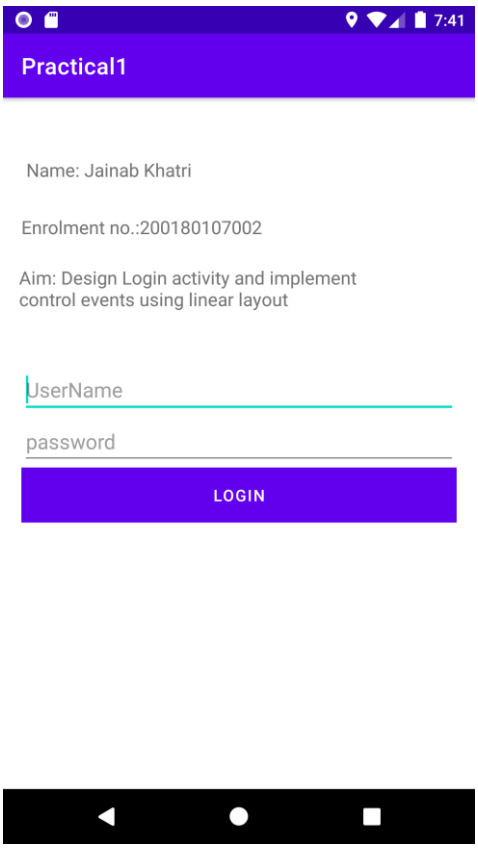
```
        android:layout_marginStart="16dp"
        android:layout_marginTop="102dp"
        android:layout_marginEnd="160dp"
        android:layout_marginBottom="278dp"
        android:text="Enrolment no.:200180107002"
        android:textSize="16sp" />
    <TextView
        android:id="@+id/textView10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="truade"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="14dp"
        android:layout_marginTop="146dp"
        android:layout_marginEnd="51dp"
        android:layout_marginBottom="278dp"
        android:text="@string/aim"
        android:textSize="16sp" />
</RelativeLayout>
```

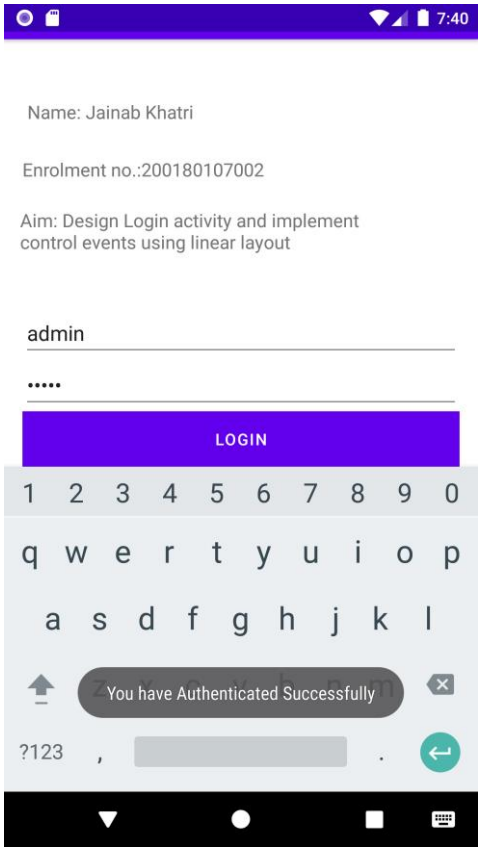
→Practical_1.java

```
package com.example.practical1;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import  java.util.Objects;
```

```
public class MainActivity extends AppCompatActivity {
    EditText username,password;
    Button login;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        username = findViewById(R.id.username);
        password = findViewById(R.id.password);
        login = findViewById(R.id.login);
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
              if(Objects.equals(username.getText().toString(),"admin")&&Objects.equals(password.getText().toString(),
"admin")) {
                Toast.makeText(MainActivity.this,"You have Authenticated Successfully",
Toast.LENGTH_LONG).show();
              } else {
                Toast.makeText(MainActivity.this, "Authentication Failed", Toast.LENGTH_LONG).show();
              }
            }
        });
    }
}
```

Output:





References used by the students:

1) https://developer.android.com/develop/ui/views/layout/declaring-layout
2) https://developer.android.com/reference/android/widget/LinearLayout
3) https://developer.android.com/reference/android/widget/TableLayout
4) https://developer.android.com/reference/android/widget/RelativeLayout

Rubric wise marks obtained:

	1	2	3	4	5	TOTAL
RUBRICS						
MARKS						

Practical-2

AIM: Create Activities & implement following

1. Implicit intent
2. Explicit Intent
3. Start Activity for Result

Code: XML FILE, JAVA FILE

→Activity_.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:background="#FFFFFF">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:orientation="vertical">
        <EditText
            android:id="@+id/username"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="UserName" />
        <EditText
            android:id="@+id/password"
            android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:hint="password"
        android:inputType="textPassword" />
    <Button
        android:id="@+id/login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#0F3CAB"
        android:text="Login"
        android:textColor="#FFFFFF" />
</LinearLayout>
<TextView
    android:id="@+id/textView8"
    android:layout_width="203dp"
    android:layout_height="29dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_marginStart="10dp"
    android:layout_marginTop="52dp"
    android:layout_marginEnd="188dp"
    android:layout_marginBottom="10dp"
    android:text="@string/name"
    android:textSize="16sp" />
<TextView
    android:id="@+id/textView9"
    android:layout_width="234dp"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
```

```
        android:layout_marginStart="16dp"
        android:layout_marginTop="102dp"
        android:layout_marginEnd="160dp"
        android:layout_marginBottom="278dp"
        android:text="@string/enrolment"
        android:textSize="16sp" />
    <TextView
        android:id="@+id/textView10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="14dp"
        android:layout_marginTop="146dp"
        android:layout_marginEnd="51dp"
        android:layout_marginBottom="278dp"
        android:text="@string/aim"
        android:textSize="16sp" />
</RelativeLayout>
```

→Activity_forgot_password.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout 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:background="#23CF0E0E">
    <TextView
        android:id="@+id/textView"
```

```
    android:layout_width="297dp"
    android:layout_height="47dp"
    android:layout_gravity="center_horizontal"
    android:background="#00000000"
    android:text="Forgot Password"
    android:textSize="34sp"
    app:layout_constraintBottom_toTopOf="@+id/emailInput"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.425"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.745" />
<EditText
    android:id="@+id/emailInput"
    android:layout_width="330dp"
    android:layout_height="24dp"
    android:hint="Enter your email"
    android:inputType="textEmailAddress"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.51"
    tools:layout_editor_absoluteX="31dp" />
<Button
    android:id="@+id/nextButton"
    android:layout_width="292dp"
    android:layout_height="49dp"
    android:background="#1C6CB2"
    android:text="Next"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
```

```
        app:layout_constraintHorizontal_bias="0.496"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/emailInput"
        app:layout_constraintVertical_bias="0.221" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

→Practical_2.java

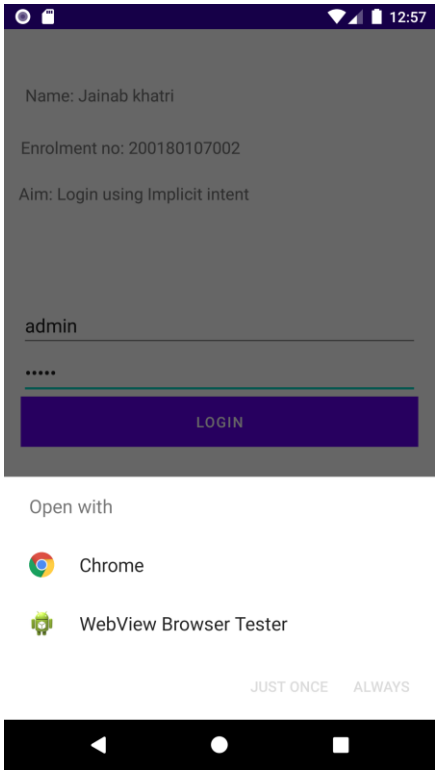
```
package com.example.practical_2;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    EditText username, password;
    Button login;
    int loginAttempts = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        username = findViewById(R.id.username);
        password = findViewById(R.id.password);
        login = findViewById(R.id.login);
        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

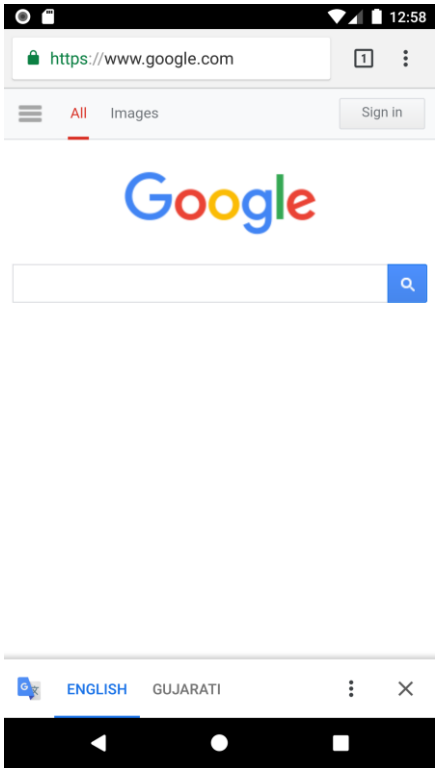
```
            if (loginAttempts < 2) {
                if (isValidCredentials(username.getText().toString(), password.getText().toString())) {
                    // Successful authentication - navigate to Google URL
                    Intent authenticationIntent = new Intent(Intent.ACTION_VIEW);
                    authenticationIntent.setData(Uri.parse("https://www.google.com/"));

                    if (authenticationIntent.resolveActivity(getPackageManager()) != null) {
                        startActivity(authenticationIntent);
                    } else {
                        Toast.makeText(MainActivity.this, "No app can handle this intent", Toast.LENGTH_SHORT).show();
                    }
                } else {
                    loginAttempts++;
                    Toast.makeText(MainActivity.this, "Authentication Failed. Attempt " + loginAttempts, Toast.LENGTH_LONG).show();
                    if(loginAttempts==2){
                        Toast.makeText(MainActivity.this, "Redirecting!!!", Toast.LENGTH_SHORT).show();
                        Intent intent = new Intent(MainActivity.this, ForgotPasswordActivity.class);
                        startActivity(intent);
                    }
                }
            }
        }
    });
  }
  private boolean isValidCredentials(String enteredUsername, String enteredPassword) {
      return enteredUsername.equals("admin") && enteredPassword.equals("admin");
  }
}
```

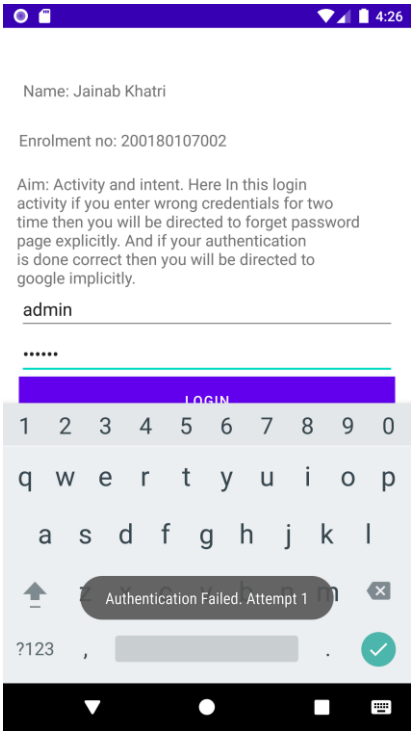
→ForgotPasswordActivity.java

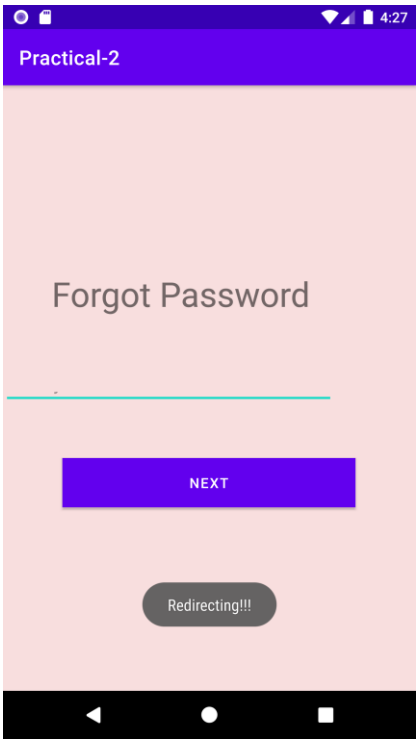
```
package com.example.practical_2;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

public class ForgotPasswordActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_forgot_password);

    }

}
```

Output:









References used by the students:

1. https://www.geeksforgeeks.org/implicit-and-explicit-intents-in-android-with-examples/
2. https://www.geeksforgeeks.org/difference-between-implicit-intent-and-explicit-intent-in-android/

Rubric wise marks obtained:

	1	2	3	4	5	TOTAL
RUBRICS						
MARKS						

Practical-3

AIM: Use an Options Menu

Code: XML FILE, JAVA FILE

→Activity_.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout 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">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Name: Jainab Khatri"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.339"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.435" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enrolment No.: 200180107002"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.315"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.484" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Aim: Use of Option Menu"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.31"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.53" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

→option_menu.xml

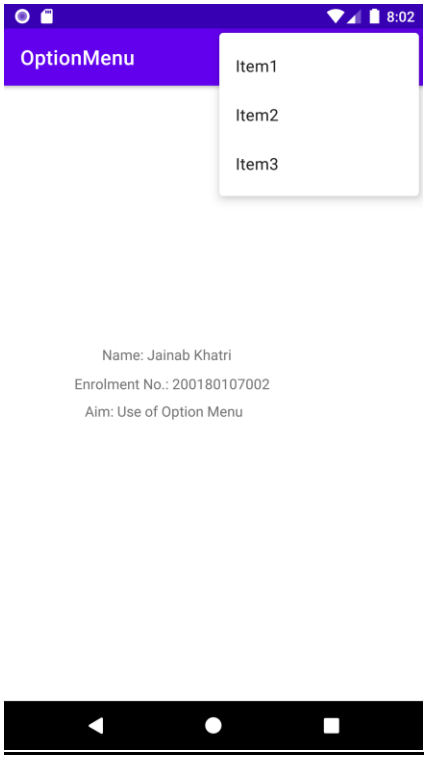
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:title="Item1"
          android:id="@+id/item1"/>
        <item android:title="Item2"
          android:id="@+id/item2"/>
        <item android:title="Item3"
          android:id="@+id/item4"/>
</menu>
```

→Practical_3.java

```
package com.example.optionmenu;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
```

```
import android.view.MenuInflater;


public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.option_menu,menu);
        // Inflate the menu; this adds items to the action bar if it is present
        return true;
    }
}
```

Output:



References used by the students:

1. https://www.geeksforgeeks.org/how-to-implement-options-menu-in-android/
2. https://developer.android.com/develop/ui/views/components/menus

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical-4

AIM: Create a Recycler View and list the details of student using following fields:

1. Name
2. Address
3. Photo (Image)
4. Delete (Button Operation)

Code: XML FILE, JAVA FILE

→Activity_.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</RelativeLayout>
```

→student_item.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="wrap_content"
    android:orientation="vertical">
    <ImageView
        android:id="@+id/studentPhoto"
        android:layout_width="100dp"
        android:layout_height="100dp"
```

```
        android:scaleType="centerCrop"
        android:src="@drawable/default_photo" />
    <TextView
        android:id="@+id/studentName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:id="@+id/studentAddress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/deleteButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Delete" />
</LinearLayout>
```

→ StudentAdapter.java

```
package com.example.recyclerview;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;
public class StudentAdapter extends RecyclerView.Adapter<StudentAdapter.ViewHolder> {
    private List<Student> students;
```

```
private OnItemClickListener listener;
public StudentAdapter(List<Student> students, OnItemClickListener listener) {
    this.students = students;
    this.listener = listener;
}
@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.student_item, parent, false);
    return new ViewHolder(view);
}
@Override
public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    Student student = students.get(position);
    holder.studentName.setText(student.getName());
    holder.studentAddress.setText(student.getAddress());
    holder.studentPhoto.setImageResource(student.getPhotoResource());
    holder.deleteButton.setOnClickListener(view -> {
        if (listener != null) {
            listener.onDeleteClick(position);
        }
    });
}
@Override
public int getItemCount() {
    return students.size();
}
public class ViewHolder extends RecyclerView.ViewHolder {
    ImageView studentPhoto;
    TextView studentName;
```

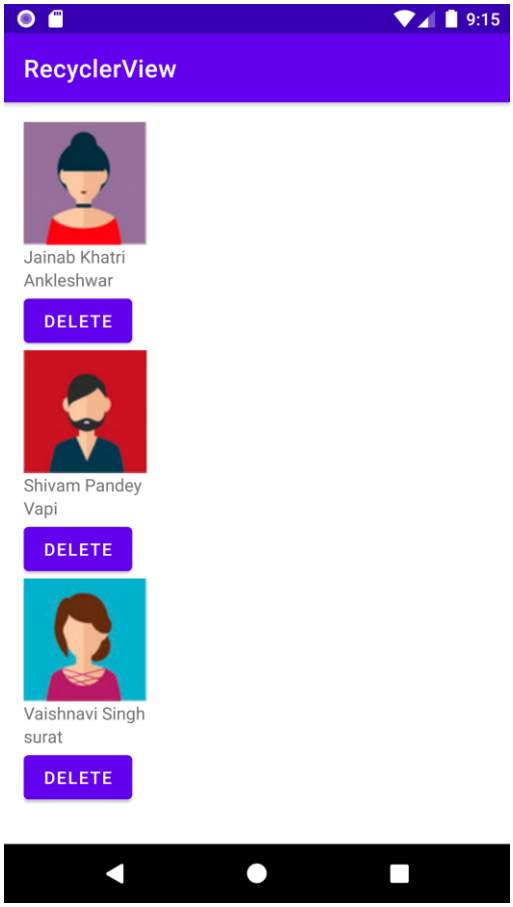
```
        TextView studentAddress;
        Button deleteButton;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            studentPhoto = itemView.findViewById(R.id.studentPhoto);
            studentName = itemView.findViewById(R.id.studentName);
            studentAddress = itemView.findViewById(R.id.studentAddress);
            deleteButton = itemView.findViewById(R.id.deleteButton);
        }
    }
    public interface OnItemClickListener {
        void onDeleteClick(int position);
    }
}
```

→Practical_4.java

```
package com.example.recyclerview;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
import java.util.List;
public class MainActivity extends AppCompatActivity {
    private RecyclerView recyclerView;
    private StudentAdapter adapter;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        recyclerView = findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        List<Student> students = generateStudentData();
        adapter = new StudentAdapter(students, position -> {
            students.remove(position);
            adapter.notifyItemRemoved(position);
        });
        recyclerView.setAdapter(adapter);
    }
    private List<Student> generateStudentData() {
        List<Student> students = new ArrayList<>();
        students.add(new Student("Jainab Khatri", "Ankleshwar", R.drawable.student2));
        students.add(new Student("Shivam Pandey", "Vapi", R.drawable.student1));
        students.add(new Student("Vaishnavi Singh", "surat", R.drawable.student3));
        return students;
    }
}
```

Output:



References used by the students:

1. https://youtu.be/TAEbP_ccjsk?si=g6RfQXJhhdVEbbNi
2. https://www.javatpoint.com/android-recyclerview-list-example

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical-5

AIM: Theme, Custom Styles, Drawable

Code: XML FILE, JAVA FILE

→Activity_.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">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">
        <TextView
            android:id="@+id/team1_text"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:text="@string/team_1"
            style="@style/TeamText" />
        <ImageButton
            android:id="@+id/decreaseTeam1"
            android:layout_width="@dimen/button_size"
            android:layout_height="@dimen/button_size"
            android:layout_alignParentStart="true"
```

```
        android:layout_centerVertical="true"
        style="@style/MinusButtons"
        android:onClick="decreaseScore" />
    <TextView
        android:id="@+id/score_1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/initial_value"
        style="@style/ScoreText" />
    <ImageButton
        android:id="@+id/increaseTeam1"
        android:layout_width="@dimen/button_size"
        android:layout_height="@dimen/button_size"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        style="@style/PlusButtons"
        android:onClick="increaseScore" />
</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1">
 <TextView
        android:id="@+id/team2_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
```

```
        android:text="@string/team_2"
        style="@style/TeamText" />
    <ImageButton
        android:id="@+id/decreaseTeam2"
        android:layout_width="@dimen/button_size"
        android:layout_height="@dimen/button_size"
        android:layout_alignParentStart="true"
        android:layout_centerVertical="true"
        style="@style/MinusButtons"
        android:onClick="decreaseScore"/>
    <TextView
        android:id="@+id/score_2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/initial_value"
        style="@style/ScoreText" />
    <ImageButton
        android:id="@+id/increaseTeam2"
        android:layout_width="@dimen/button_size"
        android:layout_height="@dimen/button_size"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        style="@style/PlusButtons"
        android:onClick="increaseScore"/>
  </RelativeLayout>
</LinearLayout>
```

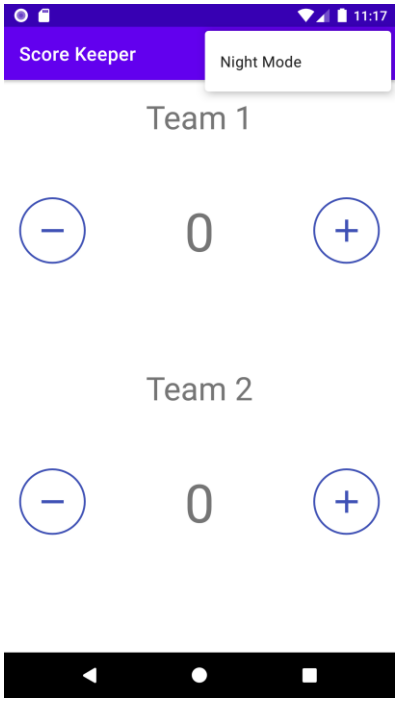
→Practical_5.java

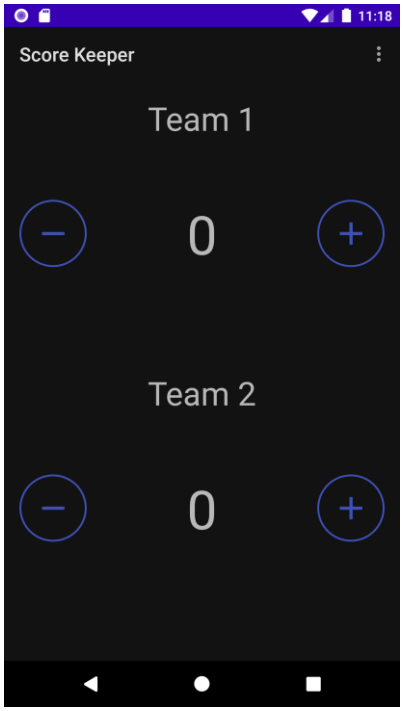
```
package com.example.scorekeeper;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatDelegate;
import android.os.Bundle;
import android.view.*;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    private int mScore1;
    private int mScore2;
    private TextView mScoreText1;
    private TextView mScoreText2;
    static final String SCORE_STATE_1 = "Team 1 score";
    static final String SCORE_STATE_2 = "Team 2 score";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mScoreText1 = findViewById(R.id.score_1);
        mScoreText2 = findViewById(R.id.score_2);
        if (savedInstanceState != null){
            mScore1 = savedInstanceState.getInt(SCORE_STATE_1);
            mScore2 = savedInstanceState.getInt(SCORE_STATE_2);
            mScoreText1.setText(String.valueOf(mScore1));
            mScoreText2.setText(String.valueOf(mScore2));
        }
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main_menu, menu);
```

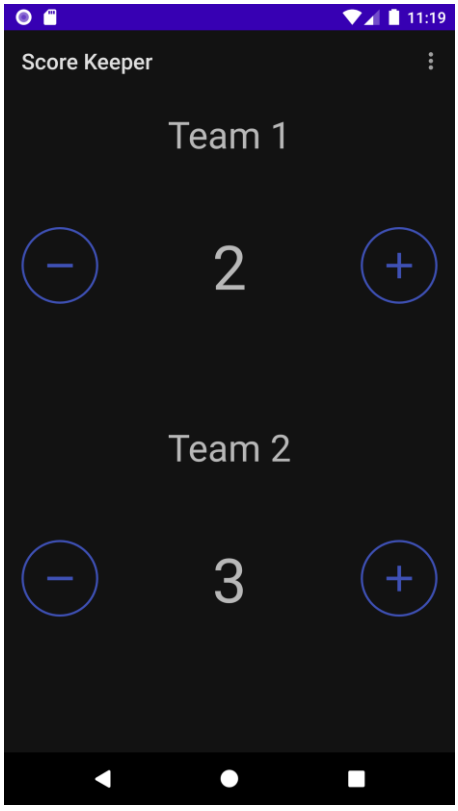
```
        int nightMode = AppCompatDelegate.getDefaultNightMode();
        if (nightMode == AppCompatDelegate.MODE_NIGHT_YES){
            menu.findItem(R.id.night_mode).setTitle(R.string.day_mode);
        }
        else{
            menu.findItem(R.id.night_mode).setTitle(R.string.night_mode);
        }
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if (item.getItemId() == R.id.night_mode){
            int nightMode = AppCompatDelegate.getDefaultNightMode();
            if (nightMode == AppCompatDelegate.MODE_NIGHT_YES){
                AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_NO);
            }
            else {
                AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_YES);
            }
            recreate();
        }
        return true;
    }
    public void decreaseScore(View view) {
        switch (view.getId()){
            case R.id.decreaseTeam1:
                if (mScore1 == 0){
                    break;
                }
                mScore1 --;
```

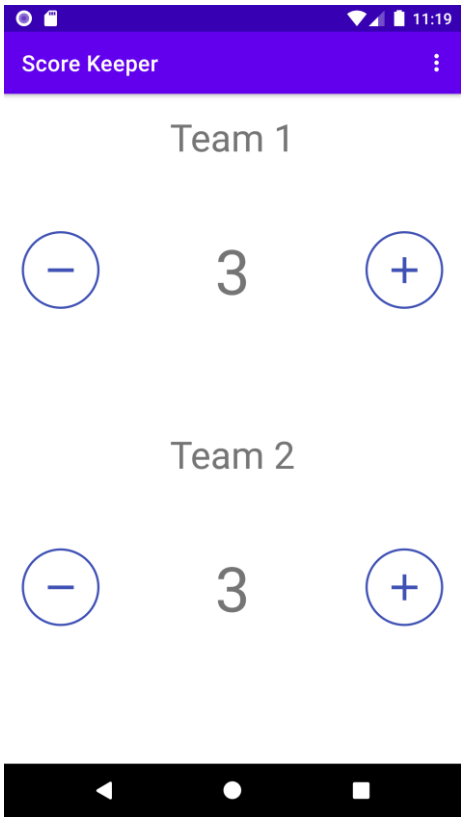
```
            mScoreText1.setText(String.valueOf(mScore1));
            break;
        case R.id.decreaseTeam2:
            if (mScore2 == 0){
                break;
            }
            mScore2 --;
            mScoreText2.setText(String.valueOf(mScore2));
            break;
    }
  }
  public void increaseScore(View view) {
    switch (view.getId()){
        case R.id.increaseTeam1:
            mScore1 ++;
            mScoreText1.setText(String.valueOf(mScore1));
            break;
        case R.id.increaseTeam2:
            mScore2 ++;
            mScoreText2.setText(String.valueOf(mScore2));
            break;
    }
  }
  @Override
  protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putInt(SCORE_STATE_1, mScore1);
    outState.putInt(SCORE_STATE_2, mScore2);
  }
}
```

Output:


11:17
Score Keeper
Night Mode
Team 1
0
Team 2
0


11:18
Score Keeper
Team 1
0
Team 2
0


11:19
Score Keeper
Team 1
2
Team 2
3


11:19
Score Keeper
Team 1
3
Team 2
3

References used by the students:

1. https://medium.com/@kreynaldi04/android-fundamentals-05-1-drawables-styles-and-themes-8b03437399b8
2. https://stackoverflow.com/questions/51331732/create-and-use-android-drawable-reference-in-your-android-theme
3. https://developer.android.com/develop/ui/views/theming/themes

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical-6

AIM: Save user data in a database

Code: XML FILE, JAVA FILE

→Activity_.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:padding="14dp"
    tools:context=".MainActivity">
    <EditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name" />
    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextName"
        android:hint="Email" />
    <EditText
        android:id="@+id/editTextEnrollmentNumber"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextEmail"
```

```
    android:hint="Enrollment Number" />
<EditText
    android:id="@+id/editTextSemester"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editTextEnrollmentNumber"
    android:hint="Semester" />
<EditText
    android:id="@+id/editTextDepartment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editTextSemester"
    android:hint="Department" />
<Button
    android:id="@+id/buttonSave"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/editTextDepartment"
    android:layout_centerHorizontal="true"
    android:text="Save" />
<Button
    android:id="@+id/buttonViewAll"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/buttonSave"
    android:layout_centerHorizontal="true"
    android:text="View All" />
<ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"
        android:layout_below="@+id/buttonViewAll"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

→Practical_6.java

```
package com.example.practical6;
import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private EditText editTextName, editTextEmail, editTextEnrollmentNumber, editTextSemester, editTextDepartment;
    private MyDatabaseHelper dbHelper;
    private SimpleCursorAdapter adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editTextName = findViewById(R.id.editTextName);
        editTextEmail = findViewById(R.id.editTextEmail);
        editTextEnrollmentNumber = findViewById(R.id.editTextEnrollmentNumber);
        editTextSemester = findViewById(R.id.editTextSemester);
```

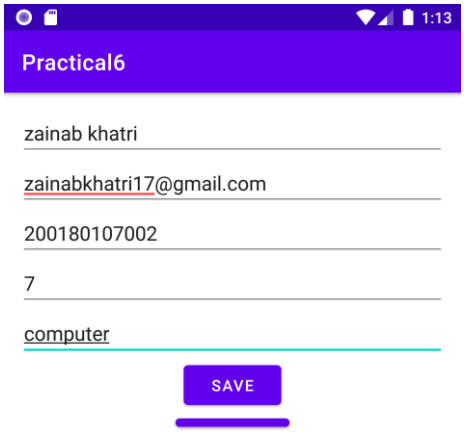
```
    editTextDepartment = findViewById(R.id.editTextDepartment);
    dbHelper = new MyDatabaseHelper(this);
    Button buttonSave = findViewById(R.id.buttonSave);
    buttonSave.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            saveUserData();
        }
    });
    Button buttonViewAll = findViewById(R.id.buttonViewAll);
    buttonViewAll.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            viewAllUserData();
        }
    })
    ListView listView = findViewById(R.id.listView);
    adapter = new SimpleCursorAdapter(
            this,
            android.R.layout.simple_list_item_2,
            null, // Cursor will be set when viewing all data
            new String[]{"_id", "name", "email", "enrollmentNumber", "semester", "department"},
            new int[]{android.R.id.text1, android.R.id.text2}
    );
    listView.setAdapter(adapter);
}
private void saveUserData() {
    SQLiteDatabase db = dbHelper.getWritableDatabase();
    String name = editTextName.getText().toString();
    String email = editTextEmail.getText().toString();
```

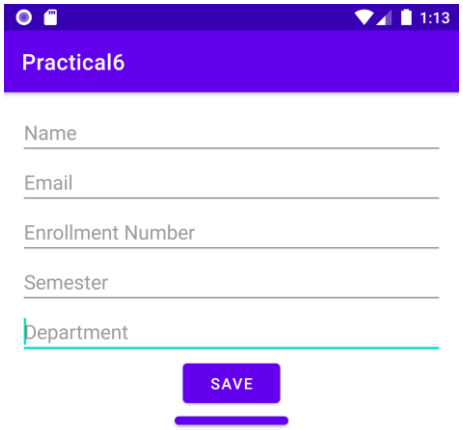
```
        String enrollmentNumber = editTextEnrollmentNumber.getText().toString();
        String semester = editTextSemester.getText().toString();
        String department = editTextDepartment.getText().toString();
        ContentValues values = new ContentValues();
        values.put("name", name);
        values.put("email", email);
        values.put("enrollmentNumber", enrollmentNumber);
        values.put("semester", semester);
        values.put("department", department);
        long newRowId = db.insert("Users", null, values);
        if (newRowId != -1) {
            Toast.makeText(this, "Data saved with ID " + newRowId, Toast.LENGTH_SHORT).show();
            editTextName.setText("");
            editTextEmail.setText("");
            editTextEnrollmentNumber.setText("");
            editTextSemester.setText("");
            editTextDepartment.setText("");
        } else {
            Toast.makeText(this, "Error saving data", Toast.LENGTH_SHORT).show();
        }

        db.close();
    }
    private void viewAllUserData() {
        SQLiteDatabase db = dbHelper.getReadableDatabase();
        String[] projection = {"_id", "name", "email", "enrollmentNumber", "semester", "department"};
        Cursor cursor = db.query("Users", projection, null, null, null, null, null);
        adapter.changeCursor(cursor);
    }
}
```

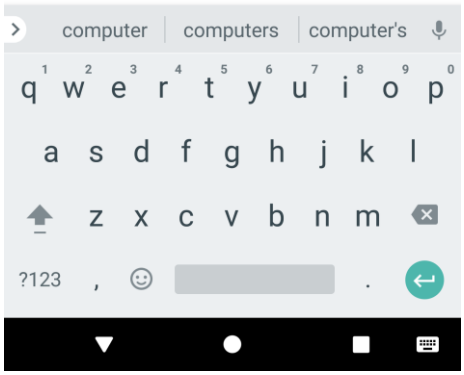
→MyDatabaseHelper.java

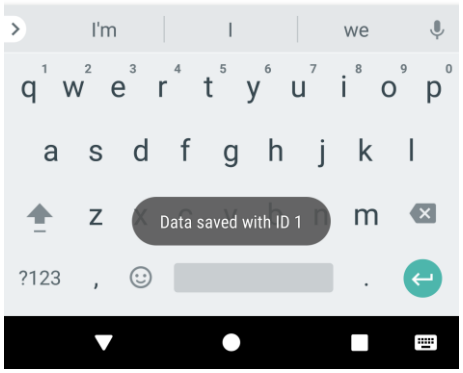
```
package com.example.practical6;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class MyDatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "UserData.db";
    private static final int DATABASE_VERSION = 2;
    public MyDatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE Users (_id INTEGER PRIMARY KEY AUTOINCREMENT, " +
                "name TEXT, email TEXT, enrollmentNumber TEXT, semester TEXT, department TEXT);");
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS Users;");
        onCreate(db);
    }
}
```

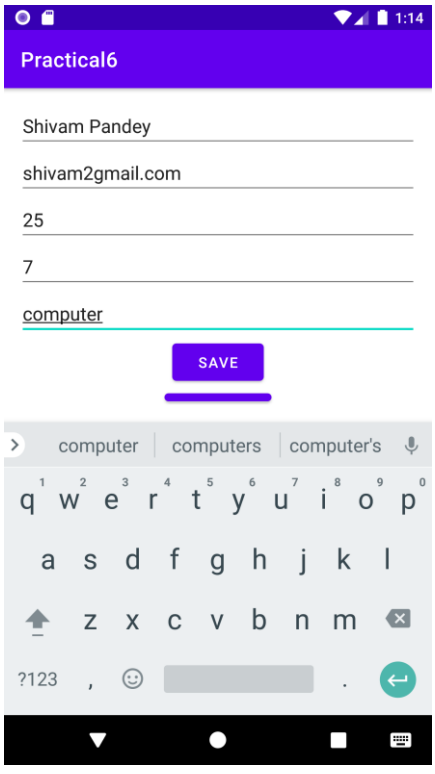
Output:

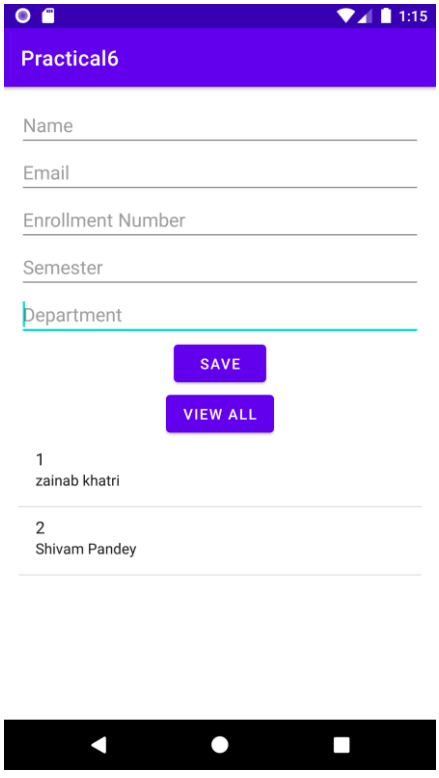

1:13
Practical6
zainab khatri
zainabkhatri17@gmail.com
200180107002
7
computer
SAVE

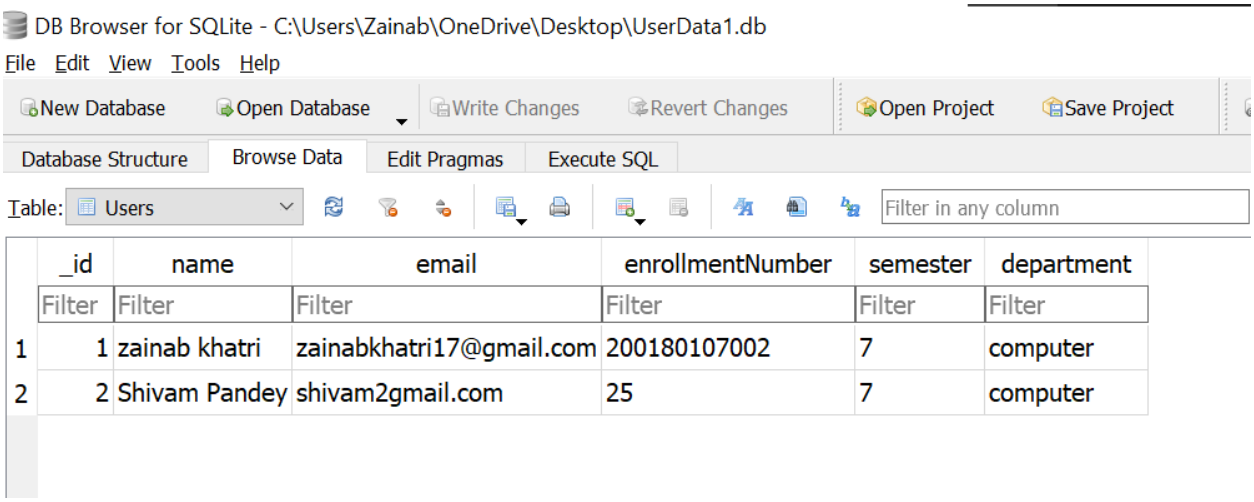

1:13
Practical6
Name
Email
Enrollment Number
Semester
Department
SAVE


computer
computers
computer's


I'm
I
we
Data saved with ID 1


1:14
Practical6
Shivam Pandey
shivam2gmail.com
25
7
computer
SAVE
computer
computers
computer's


1:15
Practical6
Name
Email
Enrollment Number
Semester
Department
SAVE
VIEW ALL
1
zainab khatri
2
Shivam Pandey



	_id	name	email	enrollmentNumber	semester	department
	Filter	Filter	Filter	Filter	Filter	Filter
1	1	zainab khatri	zainabkhatri17@gmail.com	200180107002	7	computer
2	2	Shivam Pandey	shivam2gmail.com	25	7	computer

References used by the students:

1. https://youtu.be/T5LY60lxBVc?si=oFofzOhArq62KJb0
2. https://developer.android.com/training/data-storage/sqlite
3. https://www.geeksforgeeks.org/how-to-create-and-add-data-to-sqlite-database-in-android/

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical-7

AIM: Get and Save User Preferences

Code: XML FILE, JAVA FILE

→Activity_.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:layout_gravity="center"
        android:layout_margin="20dp"
        android:gravity="center"
        android:text="Aim: Get and save data in android using Shared Preference."
        android:textColor="#000000"
        android:textSize="18dp"
        android:textStyle="bold" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="20dp"
```

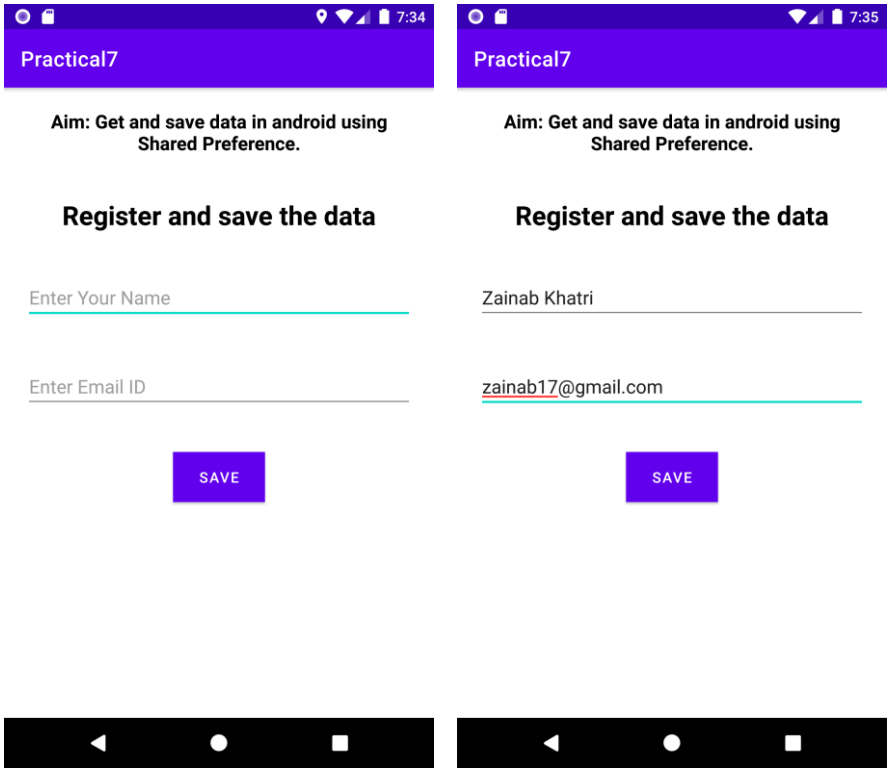
```
    android:gravity="center"
    android:text="Register and save the data"
    android:textColor="#000000"
    android:textSize="25dp"
    android:textStyle="bold" />
  <EditText
    android:id="@+id/editext_name"
    android:layout_margin="20dp"
    android:hint="Enter Your Name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
  </EditText>
  <EditText
    android:id="@+id/editext_email"
    android:layout_margin="20dp"
    android:hint="Enter Email ID"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
  </EditText>
  <Button
    android:id="@+id/button_save"
    android:textColor="#fff"
    android:background="#388E3C"
    android:layout_margin="20dp"
    android:layout_gravity="center"
    android:text="Save"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
  </Button>
</LinearLayout>
```

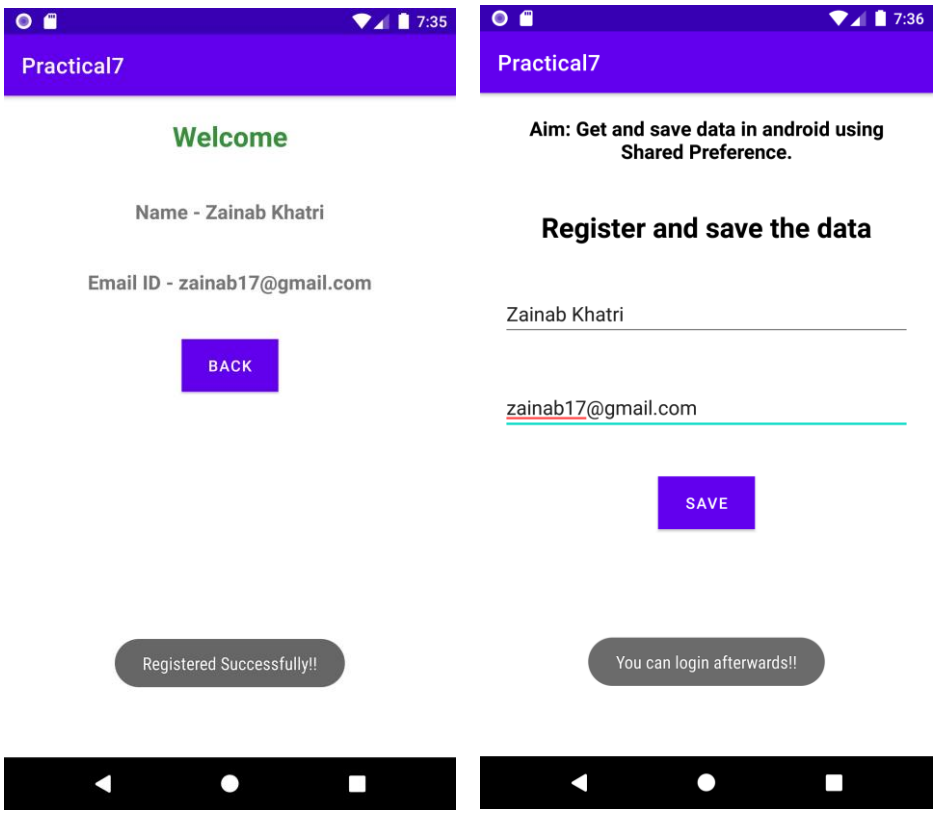
→Practical_7.java

```
package com.example.practical7;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    EditText editText_name,editText_email;
    Button button_save;
    SharedPreferences sharedPreferences;
    private static final String SHARED_PREF_NAME = "mypref";
    private static final String KEY_NAME = "name";
    private static final String KEY_EMAIL = "email";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editText_name = findViewById(R.id.editext_name);
        editText_email = findViewById(R.id.editext_email);
        button_save = findViewById(R.id.button_save);
        sharedPreferences = getSharedPreferences(SHARED_PREF_NAME,MODE_PRIVATE);
        String name = sharedPreferences.getString(KEY_NAME,null);
        if (name != null){
            Intent intent = new Intent(MainActivity.this,HomeActivity.class);
            startActivity(intent);
        }
```

```
        button_save.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //when click a button put data on Shared preferences..
                SharedPreferences.Editor editor =sharedPreferences.edit();
                editor.putString(KEY_NAME,editText_name.getText().toString());
                editor.putString(KEY_EMAIL,editText_email.getText().toString());
                editor.apply();
                Intent intent = new Intent(MainActivity.this,HomeActivity.class);
                startActivity(intent);
                Toast.makeText(MainActivity.this, "Registered Successfully!!", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Output:





References used by the students:

1. https://developer.android.com/training/data-storage/shared-preferences
2. https://www.geeksforgeeks.org/shared-preferences-in-android-with-examples/
3. https://youtu.be/pO70tQ2kgoo?si=vrDSDprXAuCCF3O8

Rubric wise marks obtained:

	1	2	3	4	5	TOTAL
RUBRICS						
MARKS						

Practical-8

AIM: Make a use of android system

Code: XML FILE, JAVA FILE

→Activity_.xml

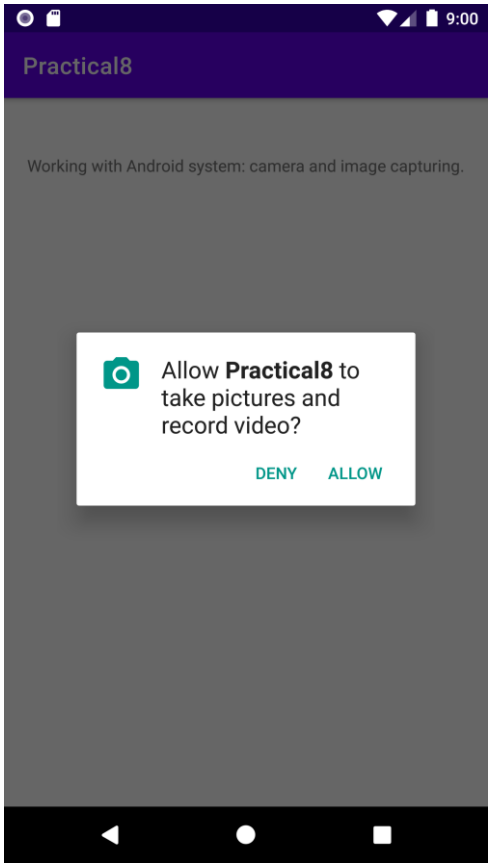
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/infoTextView"
        android:text="Working with Android system: camera and image capturing."
        android:layout_centerHorizontal="true"
        android:layout_marginTop="32dp" />
    <Button
        android:id="@+id/captureButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/infoTextView"
        android:layout_alignParentStart="true"
        android:layout_marginStart="104dp"
        android:layout_marginTop="215dp"
        android:text="Capture Image" />
</RelativeLayout>
```

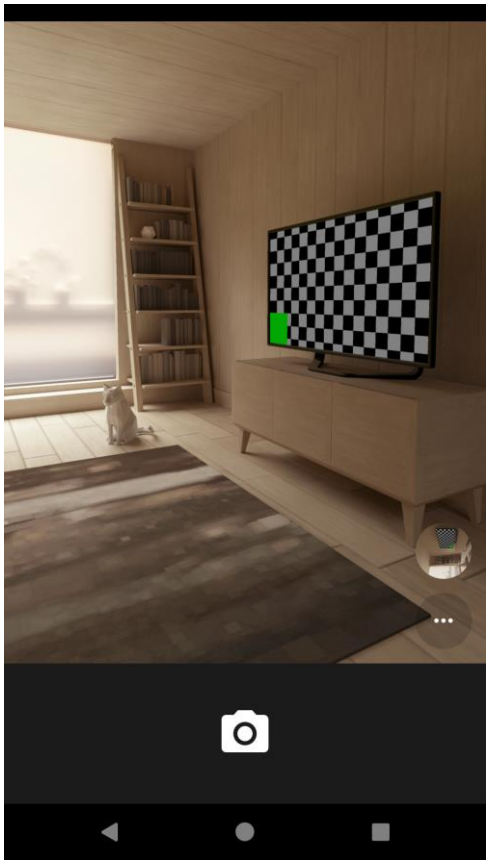
→Practical_8.java

```
package com.example.practical8;

import android.Manifest;

import android.content.pm.PackageManager;

import android.os.Bundle;

import android.content.Intent;

import android.provider.MediaStore;

import android.view.View;

import android.widget.Button;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {

    private static final int CAMERA_PERMISSION_REQUEST = 1;

    private static final int CAMERA_REQUEST = 2;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        Button captureButton = findViewById(R.id.captureButton);

        captureButton.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                if (ContextCompat.checkSelfPermission(MainActivity.this, Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {

                    ActivityCompat.requestPermissions(MainActivity.this, new String[]{Manifest.permission.CAMERA}, CAMERA_PERMISSION_REQUEST);

                } else {

                    // Permission is already granted, open the camera
```

```
                openCamera();
            }
        }
    });
}
private void openCamera() {
    Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    startActivityForResult(cameraIntent, CAMERA_REQUEST);
}
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == CAMERA_PERMISSION_REQUEST) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            // Camera permission granted, open the camera
            openCamera();
        }
    }
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == CAMERA_REQUEST && resultCode == RESULT_OK) {
        // Handle the captured image, e.g., save or display it
    }
}
}
```

Output:





References used by the students:

1. https://developer.android.com/reference/android/graphics/Camera
2. https://youtu.be/59taMJThsFU?si=HlUGc-f4Vp0QBITt

Rubric wise marks obtained:

	1	2	3	4	5	TOTAL
RUBRICS						
MARKS						

Practical-9

AIM: Using location service get the current location and display in Text View.

Code: XML FILE, JAVA FILE

→Activity_.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:orientation="vertical"
    android:layout_gravity="center"
    android:gravity="center"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/locationTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Location: "
        android:textSize="18sp"
        android:textStyle="bold"
        android:layout_marginTop="16dp"/>
</LinearLayout>
```

→Practical_9.java

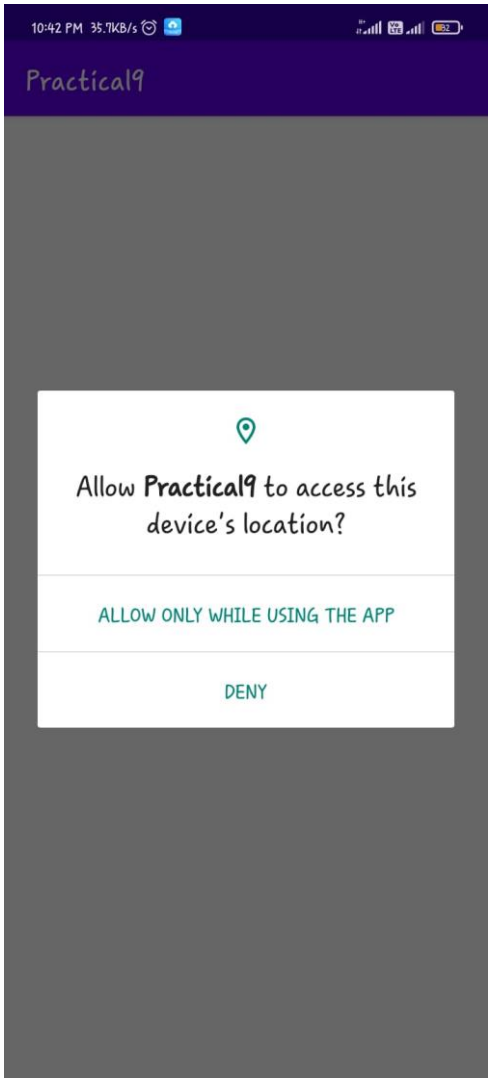
```
package com.example.practical9;
import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Address;
```

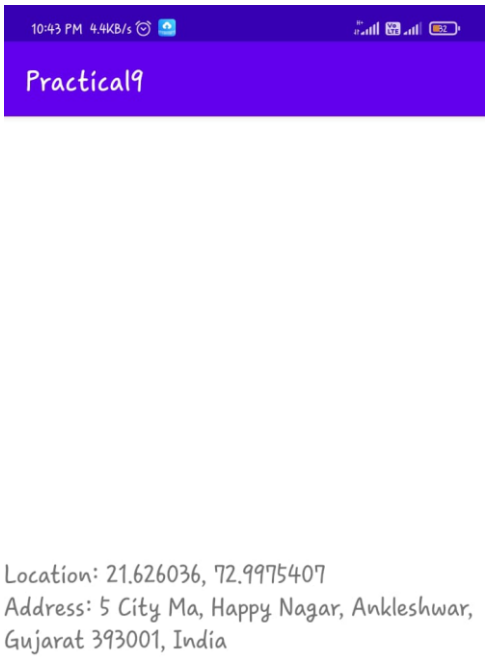
```
import android.location.Geocoder;
import android.location.Location;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;
import java.io.IOException;
import java.util.List;
import java.util.Locale;
public class MainActivity extends AppCompatActivity {
    private static final int LOCATION_PERMISSION_REQUEST = 1;
    FusedLocationProviderClient fusedLocationProviderClient;
    private TextView locationTextView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        locationTextView = findViewById(R.id.locationTextView);
        fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this);
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
LOCATION_PERMISSION_REQUEST);
        } else {
```

```
            getLocation();
        }
    }
    private void getLocation() {
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
                == PackageManager.PERMISSION_GRANTED) {
            fusedLocationProviderClient.getLastLocation().addOnSuccessListener(this, new
OnSuccessListener<Location>() {
                @Override
                public void onSuccess(Location location) {
                    if (location != null) {
                        double latitude = location.getLatitude();
                        double longitude = location.getLongitude();
                        locationTextView.setText("Location: " + latitude + ", " + longitude);
                        Geocoder geocoder = new Geocoder(MainActivity.this, Locale.getDefault());
                        try {
                            List<Address> addresses = geocoder.getFromLocation(latitude, longitude, 1);
                            if (addresses != null && !((List<?>) addresses).isEmpty()) {
                                String address = addresses.get(0).getAddressLine(0);
                                locationTextView.append("\nAddress: " + address);
                            }
                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                    } else {
                        locationTextView.setText("Location: Not available");
                    }
                }
            });
        } else {
```

```
            Toast.makeText(this, "Location permission denied.", Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults); // Call super first
        if (requestCode == LOCATION_PERMISSION_REQUEST) {
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                getLocation();
            } else {
                Toast.makeText(this, "Location permission denied.", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

Output:





References used by the students:

1. https://droidbyme.medium.com/get-current-location-using-fusedlocationproviderclient-in-android-cb7ebf5ab88e
2. https://developer.android.com/training/location/retrieve-current

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical-10

AIM: Display the use of animations

Code: XML FILE, JAVA FILE

→Activity_.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:padding="16dp"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/animateButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Animate Me"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

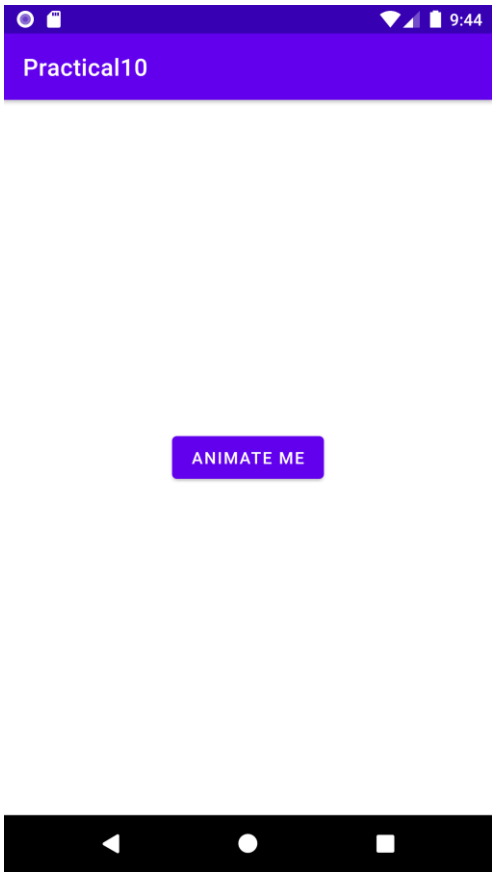
→Fade_in.xml

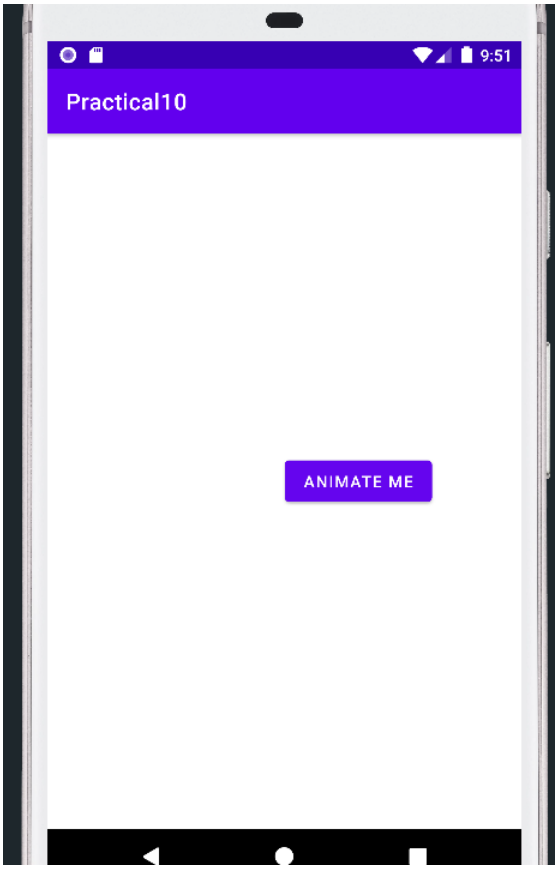
```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:fromXDelta="0%"
        android:toXDelta="100%"
        android:duration="1000" />
</set>
```

→Practical_10.java

```
package com.example.practical10;

import android.os.Bundle;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final Button animateButton = findViewById(R.id.animateButton);
        Animation fadeAnimation = AnimationUtils.loadAnimation(this, R.anim.fade_in);
        animateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                animateButton.startAnimation(fadeAnimation);
            }
        });
    }
}
```

Output:





References used by the students:

1. https://developer.android.com/develop/ui/views/animations/overview
2. https://www.geeksforgeeks.org/animation-in-android-with-example/

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						