Laboratory Manual for

Mobile Application Development (3170726)

B.E. Semester 7
Computer Engineering

Government Engineering College, Bhavnagar

Student Name	Enrollment Number
Dhruv Ambaliya	210210107508

Government Engineering College, Bhavnagar

Certificate

This is to certify that Mr. _____
Enrollment No. _____ of B.E. Semester 7th, Computer Engineering Department of Government Engineering College, Bhavnagar (GTU Code: 021) has satisfactorily completed the Practical work for the subject Mobile Application Development (3170726) for the academic year 2023-24.

Place: _____

Date: _____

Name and Sign of Faculty member **Head of the Department**

Preface

With the rapid growth of internet users over the globe, the rate of cybercrime is also increasing. Nowadays, Internet applications become an essential part of every discipline with their variety of domain-specific applications. The basic objectives to offer this course is to aware engineering graduates to understand cybercrimes and their Operandi to analyze the attack.

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.

Index
(Progressive Assessment Sheet)

Sr. No.	Objective(s) of Experiment	Page No.	Date of performance	Date of submission	Assessment Marks	Sign. of Faculty with date	Remarks
0	Write the Following 1. Vision & Mission of DTE, GEC Bhavnagar and Computer Engineering Department 2. Program Outcome of Computer Engineering 3. PSOs and PEOs of Computer Engineering Department 4. Course outcomes Mobile Application Development						
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						

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

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    EditText emailEditText,passwordEditText;
    CheckBox checkBox;
    Button loginButton;
    TextView emailTextView,passwordTextView,userDetailsTextView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        emailEditText = findViewById(R.id.edtEmail);
        passwordEditText = findViewById(R.id.edtPass);
        checkBox = findViewById(R.id.checkBox);
        loginButton = findViewById(R.id.button);
        emailTextView = findViewById(R.id.txtEmail);
        passwordTextView = findViewById(R.id.txtPassword);

        checkBox.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton compoundButton, boolean
isChecked) {
                if (isChecked) {
                    passwordEditText.setInputType(InputType.TYPE_CLASS_TEXT);
                } else {
                    passwordEditText.setInputType(InputType.TYPE_CLASS_TEXT |
InputType.TYPE_TEXT_VARIATION_PASSWORD);
                }
                passwordEditText.setSelection(passwordEditText.length());
            }
        });

        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                59

                emailTextView.setText("Email: " + email);
                passwordTextView.setText("Password: " + password);
            }
```

```
            });
        }
    }
```

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">

    <TextView
        android:id="@+id/textView"
        ../>
    <TextView
        android:id="@+id/textView2"
        ...
        android:text="EN.NO  210210107508"
        ... />

    <EditText
        android:id="@+id/edtEmail"
        ...
        android:hint="Email"
        android:inputType="text"
        />
<EditText
            android:id="@+id/edtPass"
            ...
            android:ems="10"
            android:hint="Password"
            android:inputType="textPassword"
            android:padding="10dp" />

        <CheckBox
            android:id="@+id/checkBox"
            ...
            android:text="show  Password"  />

        <Button
            android:id="@+id/button"
            ...
            android:text="Submit"  />
        <TextView
            android:id="@+id/txtEmail"
            ... />
        <TextView
```

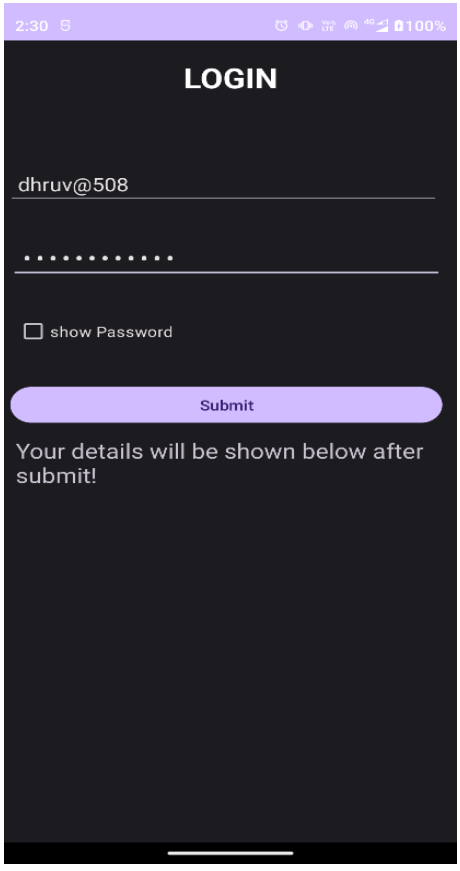
```
            android:id="@+id/txtPassword"
            ... />


</LinearLayout>
```

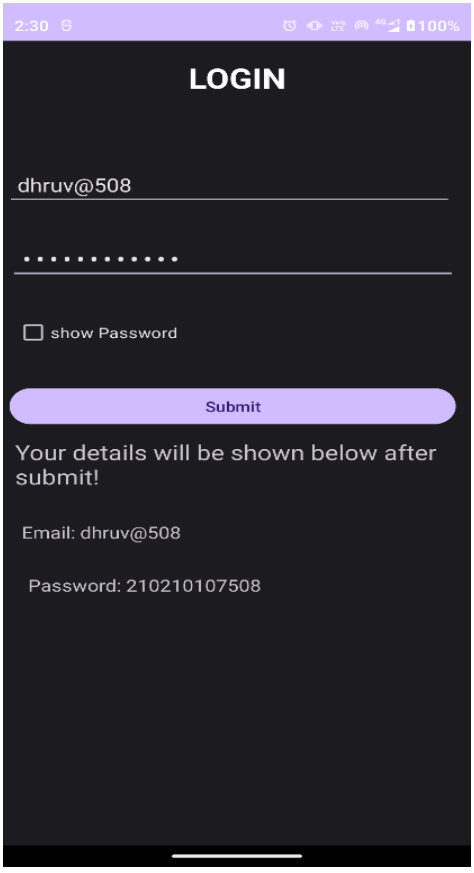
Relative_activity.xml

```
<?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"
  tools:context=".RelativeLoginScreen">

  <TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    ... />

  <TextView
    android:id="@+id/textView2"
    ... />

  <EditText
    android:id="@+id/edtEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    ...
    android:padding="10dp" />

  <EditText
    android:id="@+id/edtPass"
    ...
    android:hint="Password"
    android:inputType="textPassword"
    android:padding="10dp" />

  <CheckBox
    android:id="@+id/checkBox"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/edtPass"
    android:layout_alignStart="@+id/edtPass"
    android:layout_alignEnd="@+id/edtPass"
```

```
    ...
    android:text="show Password" />

  <Button
    android:id="@+id/button"
    android:layout_height="wrap_content"
    ...
    android:text="Submit" />
  <TextView
    android:id="@+id/txtEmail"
    ... />
  <TextView
    android:id="@+id/txtPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    ... />
</RelativeLayout>
```

Output Screenshots:





References used by the students:

https://developer.android.com/docs
https://stackoverflow.com/
https://www.geeksforgeeks.org/

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical – 2

Aim: Create Activities & implement following

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

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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

        Button urlGoBtn = (Button ) findViewById(R.id.urlGoBtn);
        EditText editTextUrl = (EditText)findViewById(R.id.editTextUrl);

        urlGoBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(Intent.ACTION_VIEW,Uri.parse(String.valueOf(editTextUrl.getText())));
                startActivity(i);
            }
        });

        Button dialBtn = (Button ) findViewById(R.id.dialBtn);
        EditText editTextPhone = (EditText)findViewById(R.id.editTextPhone);

        dialBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(Intent.ACTION_DIAL,Uri.parse("tel: "+ editTextPhone.getText()));
                startActivity(i);
            }
        });



        Button changeTextBtn=(Button) findViewById(R.id.changeTitleBtn);

        changeTextBtn.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
        public void onClick(View v) {
            Log.d("h","hm");
            Intent changeTitleIntent = new Intent(MainActivity.this,CHANGE_TITLE_ACTIVITY.class);
            startActivity(changeTitleIntent);
        }
    });
  }
}
```

CHANGE_TITLE_ACTIVITY.java

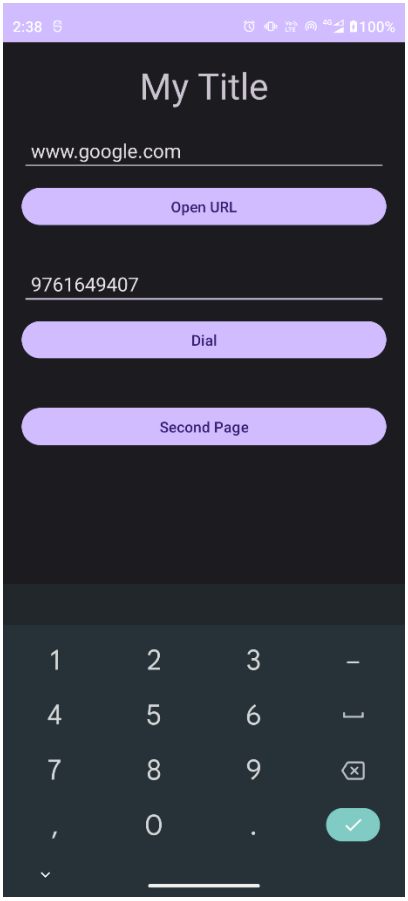
```
public class CHANGE_TITLE_ACTIVITY extends AppCompatActivity {

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

    TextView titleText = (TextView) findViewById(R.id.textView2);

    EditText editTitleText = (EditText) findViewById(R.id.editTitleText);

    Button submitTitleChangeBtn = (Button ) findViewById(R.id.submitChangeTitle);

    submitTitleChangeBtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {

        titleText.setText(editTitleText.getText());
      }
    });

  }
}
```

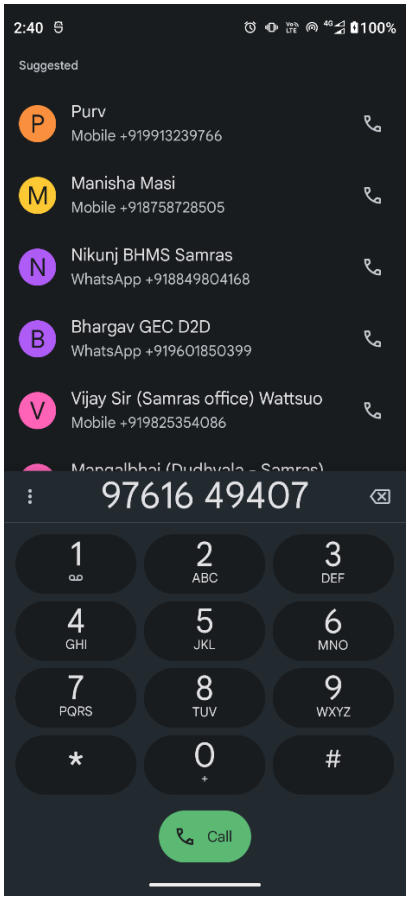
activity_main.xml

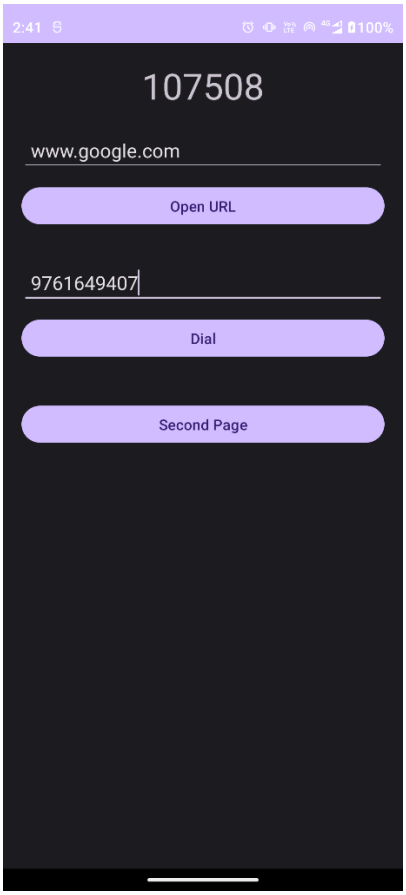
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"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

  <TextView
    android:id="@+id/textView6"
    ... />

  <TextView
    android:id="@+id/titleText"
    .../>
```

```
        <EditText
            android:id="@+id/editTextUrl"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPersonName"
            ... />
        <Button
            android:id="@+id/urlGoBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="open URL"
            ..." />
        <EditText
            android:id="@+id/editTextPhone"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="phone"
            ... />
        <TextView
            android:id="@+id/textView3"
            android:layout_width="254dp"
            android:layout_height="34dp"
            android:text="ENTER PHONE NUMBER BELOW"
            .../>
        <Button
            android:id="@+id/dialBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="DIAL"
            .../>
        <Button
            android:id="@+id/changeTitleBtn"
            android:layout_width="248dp"
            android:layout_height="48dp"
            android:text="seconde page"
            ... />
    </androidx.constraintlayout.widget.ConstraintLayout>
```

Output Screenshots:







References used by the students:

https://developer.android.com/docs

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical – 3

Aim: Use an Options Menu.

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu,menu);
        return true;
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState, @Nullable PersistableBundle persistentState) {
        super.onCreate(savedInstanceState, persistentState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if(item.getItemId()==R.id.download){
            Toast.makeText(this, "Download clicked", Toast.LENGTH_SHORT).show();
        }else if(item.getItemId()==R.id.profile){
            Toast.makeText(this, "210210107508", Toast.LENGTH_SHORT).show();
        }

        return true; // Indicates that the item has been handled

    }

}
```

activity_main.xml

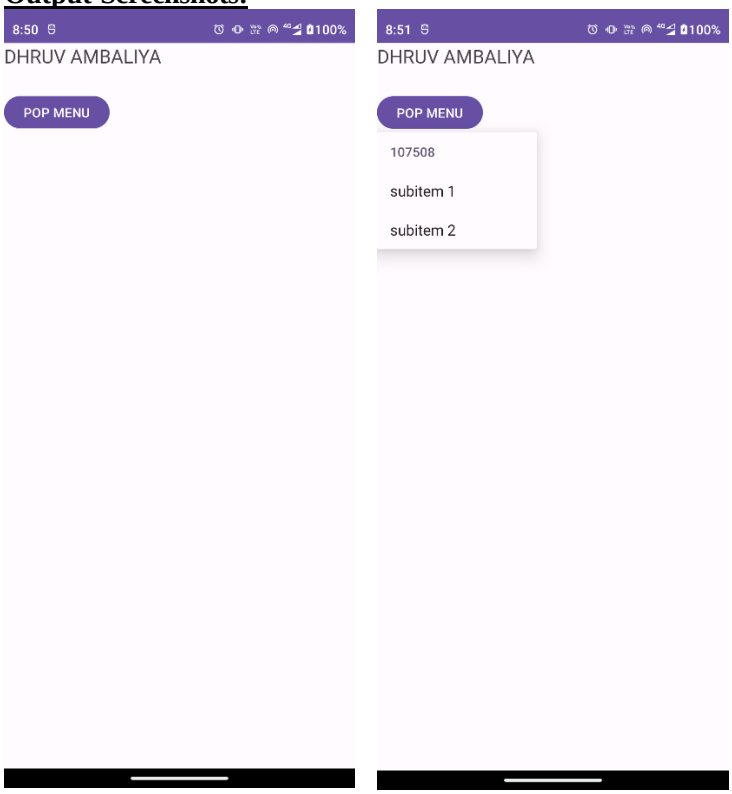
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="pop"
        android:text="Pop Menu"
        android:textAllCaps="true" />
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="DHRUV AMBALIYA"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

Menu.xml

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

    <item
        android:title="107508"
        android:id="@+id/subitem1"
        app:iconTint="@color/white"
        />
    <item
        android:title="Download"
        android:id="@+id/subitem2"
        app:iconTint="@color/white"
        />
</menu>
```

Output Screenshots:



References used by the students:

https://developer.android.com/docs
https://stackoverflow.com/
https://www.geeksforgeeks.org/

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)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
   RecyclerView recyclerView;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      try {


      recyclerView = findViewById(R.id.recyclerView);

      List<ModelClass> list = new ArrayList<>();

           List<ModelClass> list = new ArrayList<>();
            list.add(new ModelClass("Dhruv","Surat", R.drawable.user));
            list.add(new ModelClass("Ram","Rampara gam", R.drawable.user));
            list.add(new ModelClass("Yash Chauhan","Surat", R.drawable.user));
            list.add(new ModelClass("Yash Rathod","Gariyadhar",  R.drawable.user));
            list.add(new ModelClass("Hardik","Bhavnagar",  R.drawable.user));
            list.add(new ModelClass("Karan","Ahmedabad",  R.drawable.user));
            list.add(new ModelClass("Sumit","Narigam",  R.drawable.user));
            list.add(new ModelClass("Shivang","Navsari",  R.drawable.user));
            list.add(new ModelClass("Vipul","Kodinar", R.drawable.user));
            list.add(new ModelClass("Tushar","Jesar",  R.drawable.user));

      CustomAdapter customAdapter = new CustomAdapter(list);
      LinearLayoutManager layoutManager = new LinearLayoutManager(getApplicationContext());
      recyclerView.setLayoutManager(layoutManager);
      recyclerView.setAdapter(customAdapter);
      }catch (Exception e){
         Log.d("success",e.getMessage());
      }
   }
}
```

CustomAdaptor.java

```
public class CustomAdapter extends RecyclerView.Adapter<CustomAdapter.ViewHolder> {
    List<ModelClass>  list;
    public CustomAdapter(List<ModelClass> list){
        this.list = list;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.activity_row,parent,false);
        ViewHolder viewHolder = new ViewHolder(view);
        return viewHolder;
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        holder.name.setText(list.get(position).getName());
        holder.contact_no.setText(list.get(position).getContact_no());
        holder.imageView.setImageResource(list.get(position).getImageId());

        holder.delete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                deleteItem(holder.getAdapterPosition());
            }
        });
    }

    @Override
    public int getItemCount() {
        return list.size();
    }

    void deleteItem(int position){
        list.remove(position);
        notifyItemRemoved(position);
        notifyItemRangeChanged(position, list.size());

    }

    public static class ViewHolder extends RecyclerView.ViewHolder{
        TextView name,contact_no;
        ImageView imageView;
        ImageView delete;
        public ViewHolder(@NonNull View itemView) {
            super(itemView);
```

```
            name = itemView.findViewById(R.id.name);
            contact_no = itemView.findViewById(R.id.address);
            imageView = itemView.findViewById(R.id.image);
            delete = itemView.findViewById(R.id.btnDelete);
        }
    }
}
```

Model.java

```
package com.example.prac4;

public class Model {
    String name,contact_no;
    int imageId;

    public ModelClass(String name,String contact_no,int imageId){
        this.name = name;
        this.contact_no = contact_no;
        this.imageId = imageId;
    }

    public String getName() {
        return name;
    }

    public String getContact_no() {
        return contact_no;
    }

    public int getImageId() {
        return imageId;
    }

}
```

activity_main.xml

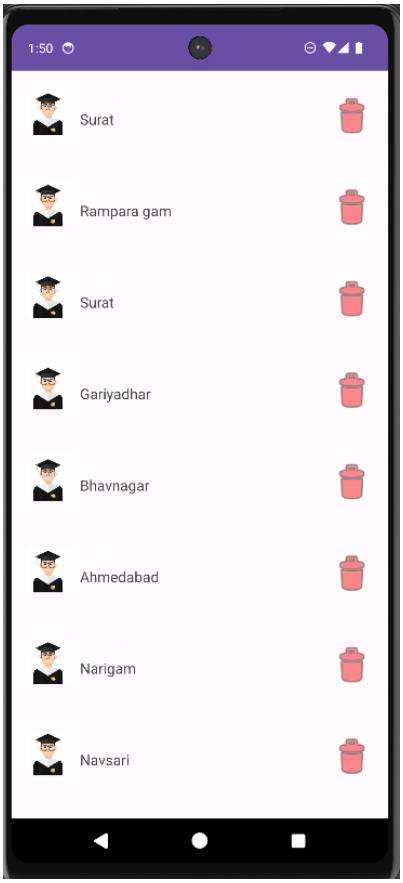
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
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

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

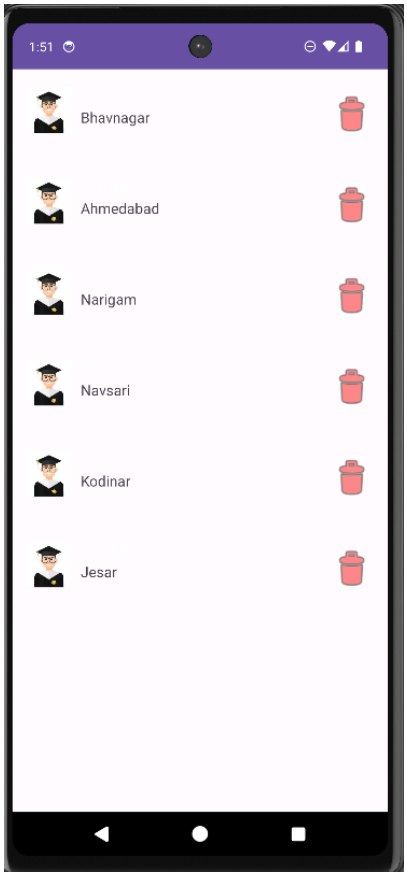
```
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

activity_row.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="wrap_content"
    android:layout_margin="15dp"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/image"
        android:layout_width="50dp"
        android:layout_height="60dp"
        android:layout_marginRight="10dp"
        android:layout_weight="0"
        tools:src="@tools:sample/avatars" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:orientation="vertical">

        <TextView
            android:id="@+id/name"
            ... />

        <TextView
            android:id="@+id/address"
            ... />
    </LinearLayout>

    <ImageView
        android:id="@+id/btnDelete"
        ...
        app:srcCompat="@android:drawable/ic_menu_delete"
        app:tint="#FB3939" />

</LinearLayout>
```

Output Screenshots:





References used by the students:

https://developer.android.com/docs
https://stackoverflow.com/

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical – 5

Aim: Theme, Custom Styles, Drawables

(Print your name, enrollment and semester on screen. Create a theme and color and use it in your application. Create a custom round button and display it on screen.)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

   Button button;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      button = findViewById(R.id.btnClick);

      button.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            Intent intent = new Intent(MainActivity.this, Detail_activity.class);
            startActivity(intent);
         }
      });

   }
}
```

Detail_activity.xml

```
public class Detail_activity extends AppCompatActivity {

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

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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <androidx.appcompat.widget.Toolbar
        android:elevation="0dp"
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/main"
        android:minHeight="?attr/actionBarSize"
        android:theme="?attr/actionBarTheme"
        app:title="@string/app_name" />

    <TextView
        ... />

    <TextView
        android:id="@+id/textView"
        ... />

    <Button
        android:id="@+id/btnClick"
        ...
        android:text="@string/button_txt"
        android:background="@color/main"/>

</LinearLayout>
```

activity_detail.xml

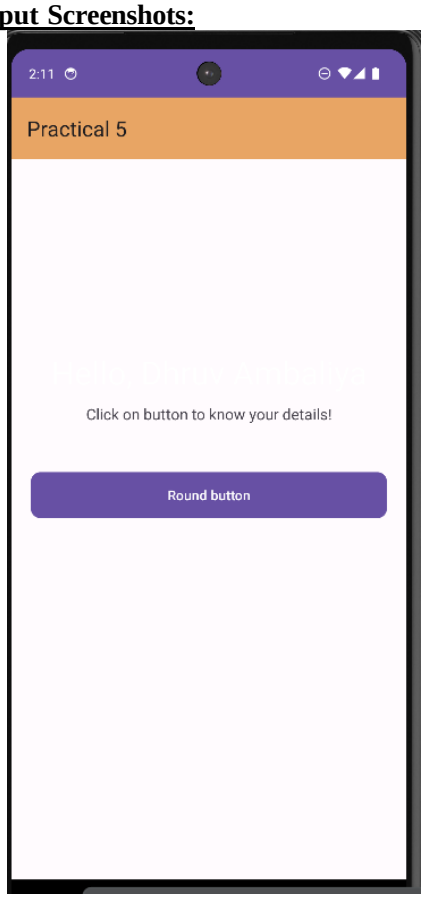
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <androidx.appcompat.widget.Toolbar
```

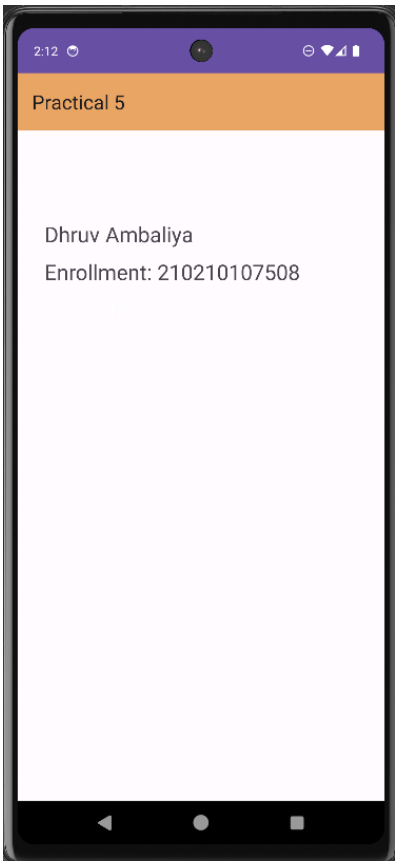
```
        android:id="@+id/toolbar2"
       ... />

    <TextView
        ...
        android:text="@string/name"
        android:textColor="@color/white"
        android:textSize="24sp"
        ... />
    <TextView
        android:id="@+id/textView"
        ... />
    <TextView
        android:id="@+id/textView3"
        ... />
</LinearLayout>
```

Theme.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.Prac5" parent="Theme.Material3.DayNight.NoActionBar">
        <!-- Customize your light theme here. -->
        <!-- <item name="colorPrimary">@color/my_light_primary</item> -->
    </style>

    <style name="Theme.Prac5" parent="Base.Theme.Prac5" />
</resources>
```

Output Screenshots:





References used by the students:

https://developer.android.com/docs
https://stackoverflow.com/
https://www.geeksforgeeks.org/

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical – 6

Aim: Practical: Save user data in a database (First Name, Last Name, Age)

MainActivity.java

```
public class MainActivity extends Activity {
   private DbHandler dbHelper;
   private SQLiteDatabase db;
   private TextView textView;

   EditText ageTextView,firstNameText,lastNameText;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      textView = findViewById(R.id.textView1);

      dbHelper = new DbHandler(this);
      db = dbHelper.getWritableDatabase();


     ageTextView = findViewById(R.id.editTextAge);
       firstNameText = findViewById(R.id.editTextFirstName);
       lastNameText = findViewById(R.id.editTextLastName);

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

      addBtn.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {

            String ageText = ageTextView.getText().toString();
            String firstName = firstNameText.getText().toString();
            String lastName = lastNameText.getText().toString();

            try {
               int age = Integer.parseInt(ageText);

               UserModel u1 = new UserModel(age, firstName, lastName);

               long newRowId = db.insert(DbHandler.TABLE_USERS, null, u1.toContentValues());

               if (newRowId != -1) {
```

```
                textView.setText("All users!\n");

                Cursor cursor = db.query(
                        DbHandler.TABLE_USERS,
                        null,
                        null,
                        null,
                        null,
                        null,
                        null
                );

                if (cursor != null) {
                    while (cursor.moveToNext()) {
                        int userAge = cursor.getInt(cursor.getColumnIndexOrThrow(DbHandler.COLUMN_AGE));
                        String userFirstName =
cursor.getString(cursor.getColumnIndexOrThrow(DbHandler.COLUMN_FIRST_NAME));
                        String userLastName =
cursor.getString(cursor.getColumnIndexOrThrow(DbHandler.COLUMN_LAST_NAME));
                        String userData = "Age: " + userAge + ", Name: " + userFirstName + " " + userLastName +
"\n";
                        textView.append(userData);
                    }
                    cursor.close();
                }
            } else {
                textView.setText("Error inserting data.");
            }
        } catch (NumberFormatException e) {
            textView.setText("Error: Please enter a valid age.");
        }
    }
});
}
}
```

DbHandler.java

```
public class DbHandler extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "user_db";
    private static final int DATABASE_VERSION = 1;

    public static final String TABLE_USERS = "users";
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_AGE = "age";
    public static final String COLUMN_FIRST_NAME = "first_name";
    public static final String COLUMN_LAST_NAME = "last_name";

    private static final String DATABASE_CREATE = "create table "
            + TABLE_USERS + "(" + COLUMN_ID
            + " integer primary key autoincrement, " + COLUMN_AGE
            + " integer, " + COLUMN_FIRST_NAME
            + " text not null, " + COLUMN_LAST_NAME
```

```
            + " text not null);";

    public DbHandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    public DbHandler(@Nullable Context context, @Nullable String name, @Nullable
SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase database) {
        database.execSQL(DATABASE_CREATE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_USERS);
        onCreate(db);
    }
}
```

Params.java

```
package com.example.prac6;

public class Params {

    public static final int DB_VERSION = 1;
    public static final String DB_NAME = "User ";
    public static final String TABLE_NAME = "User";


    public static final String KEY_ID = "id ";
    public static final String KEY_Firstname = "firstname ";
    public static final String KEY_Lastname = "lastname ";

    public static final String KEY_age = "age ";
}
```

UserModel.java

```
package com.example.prac6;
import android.content.ContentValues;
import android.text.Editable;

public class UserModel {
    private int age;
    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
```

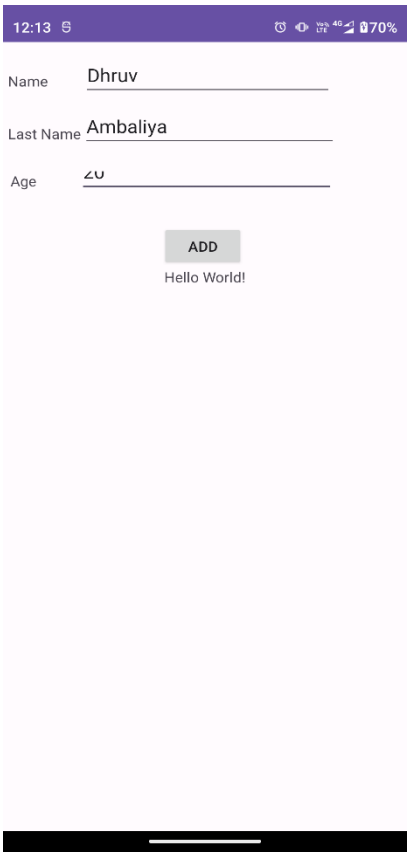
```
    public String getFirstname() {
        return Firstname;
    }

    public void setFirstname(String firstname) {
        Firstname = firstname;
    }

    public String getLastname() {
        return Lastname;
    }

    public void setLastname(String lastname) {
        Lastname = lastname;
    }

    public UserModel(int age, Editable text, Editable lastNameTextText){

    }
    public UserModel(int age, String firstname, String lastname) {
        this.age = age;
        Firstname = firstname;
        Lastname = lastname;
    }

    private  String Firstname;
    private String Lastname ;

    public ContentValues toContentValues() {
        ContentValues values = new ContentValues();
        values.put(DbHandler.COLUMN_AGE, age);
        values.put(DbHandler.COLUMN_FIRST_NAME, Firstname);
        values.put(DbHandler.COLUMN_LAST_NAME, Lastname);
        return values;
    }

}
```

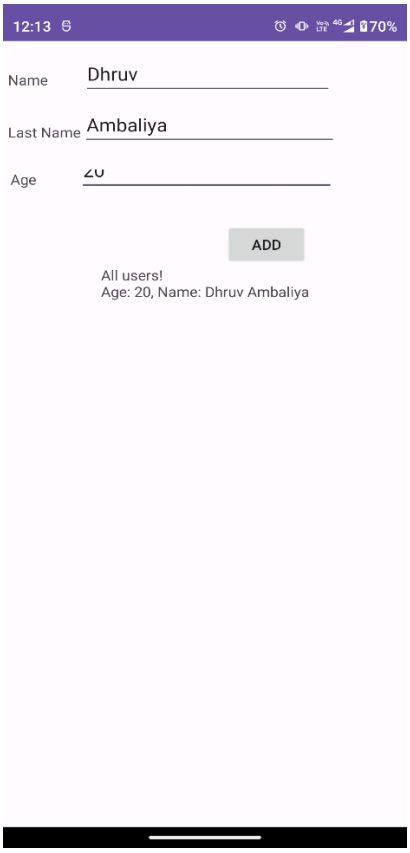
activiry_main.xml

```
<?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">

  <!-- First Name EditText -->
  <EditText
    android:id="@+id/editTextFirstName"
    ... />
  <EditText
    android:id="@+id/editTextLastName"
    .../>
  <EditText
    android:id="@+id/editTextAge"
```

```
    ... />
  <TextView
    android:id="@+id/textView1"
    ... />
  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add"
    .../>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output Screenshots:





References used by the students:

https://developer.android.com/docs

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical – 7

Aim: Get and Save User Preferences

(Get a value as input from user and save it using SharedPreferences. Display the same value next time when user launches application. If user enters new value, it should override old value. Test this application using your enrollment no. as input)

MainActivity.java

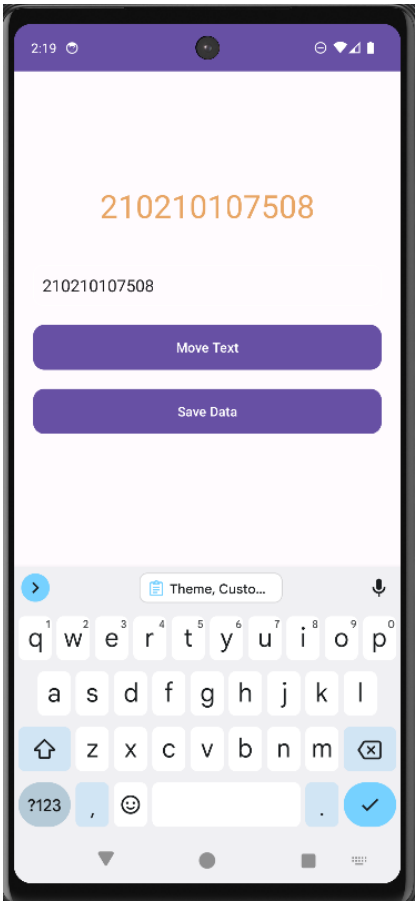
```
public class MainActivity extends AppCompatActivity {

    TextView displayText;
    Button updateButton;
    Button saveButton;
    EditText inputText;

    String storedText;

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

        displayText = findViewById(R.id.txtDisplay);
        updateButton = findViewById(R.id.btnUpdate);
        inputText = findViewById(R.id.editInput);
        saveButton = findViewById(R.id.btnSave);

        updateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                storedText = inputText.getText().toString();
                displayText.setText(storedText);
            }
        });

        saveButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                saveText();
            }
```

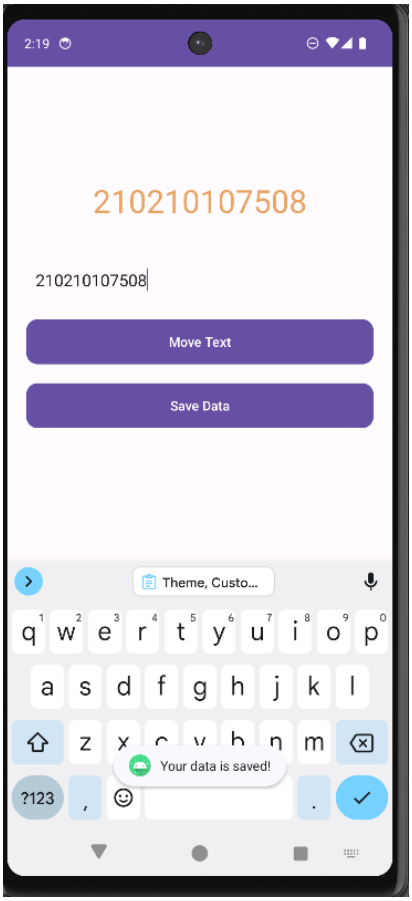
```
        });
        loadText();
    }

    public void saveText() {
        SharedPreferences sharedPreferences = getSharedPreferences("SavedText", MODE_PRIVATE);
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString("textValue", storedText);
        editor.apply();
        Toast.makeText(this, "Your data has been saved!", Toast.LENGTH_SHORT).show();
    }

    public void loadText() {
        String loadedText;
        SharedPreferences sharedPreferences = getSharedPreferences("SavedText", MODE_PRIVATE);
        loadedText = sharedPreferences.getString("textValue", "");
        displayText.setText(loadedText);
    }
}
```

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"
    android:layout_margin="20dp"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txtDisplay"
        ...

        />

    <EditText
        android:id="@+id/editInput"
        ... />

    <Button
        android:id="@+id/btnUpdate"
        ...
        android:text="Show Title"
        android:textColor="@color/white" />

    <Button
        android:id="@+id/btnSave"
        ... />

</LinearLayout>
```

Output Screenshots:





References used by the students:

https://developer.android.com/docs
https://stackoverflow.com/

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical – 8

Aim: make a use of android system

(Develop an application to click a photo using camera when user clicks on button and display the captured picture on the screen. Check and ask for the camera permission from the user.)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
    Button captureButton;
    ImageView resultImageView;

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

        captureButton = findViewById(R.id.btnCapture);
        resultImageView = findViewById(R.id.imageView);

        captureButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (checkSelfPermission(Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
                    requestCameraPermission();
                } else {
                    openCamera();
                }
            }
        });
    }

    private void requestCameraPermission() {
        requestPermissions(new String[]{Manifest.permission.CAMERA}, 123);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[]
grantResults) {
```

```
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == 123) {
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                openCamera();
            } else {
                requestCameraPermission();
            }
        }
    }

    private void openCamera() {
        Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(cameraIntent, 111);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == 111) {
            if (data != null) {
                Bitmap capturedPhoto = (Bitmap) data.getExtras().get("data");
                resultImageView.setImageBitmap(capturedPhoto);
            } else {
                Toast.makeText(this, "No image selected", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

activity_main.xml

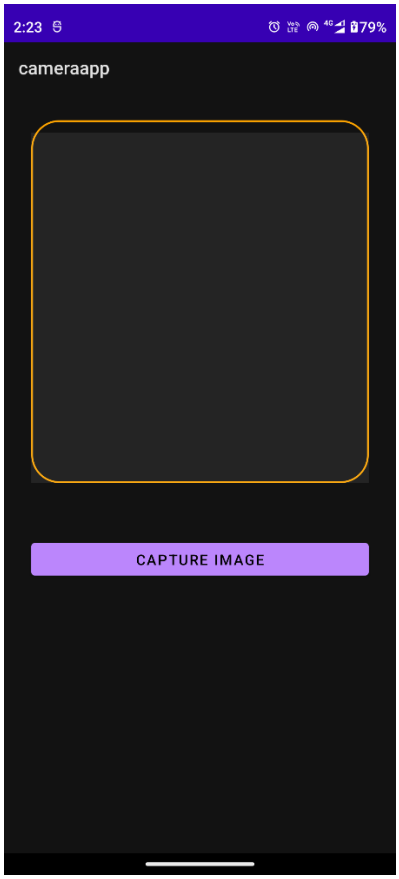
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <Button
        android:id="@+id/btnCapture"
       ...
        android:text="Capture Now" />

    <TextView
        android:id="@+id/textView"
        ...
```

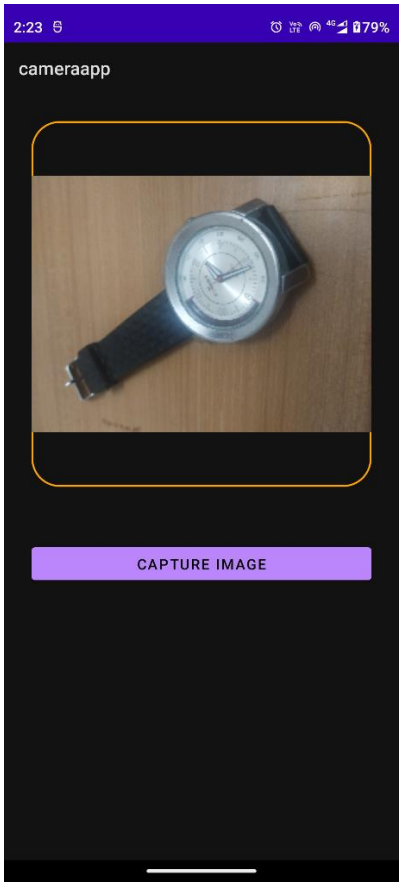
```
        android:text="210210107508" />
</LinearLayout>
```

AndroidMenifest.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">
    <uses-feature
        android:name="android.hardware.camera"
        android:required="false" />
    <uses-permission  android:name="android.permission.CAMERA"/>
```

Output Screenshots:





References used by the students:

https://developer.android.com/docs

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical – 9

Aim: Using location service get the current location and display the latitude and longitude in a TextView.

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    private static final int LOCATION_PERMISSION_REQUEST_CODE = 1;

    private TextView locationTextView;
    private FusedLocationProviderClient fusedLocationClient;

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

        locationTextView = findViewById(R.id.locationTextView);
        fusedLocationClient = LocationServices.getFusedLocationProviderClient(this);

        // Check for location permission and request it if necessary.
        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_CODE);
        } else {
            // Permission already granted, so request location updates.
            requestLocationUpdates();
        }
    }

    private void requestLocationUpdates() {
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        fusedLocationClient.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("Latitude: " + latitude + "\nLongitude: " + longitude);
```

```
            } else {
                locationTextView.setText("Location not available");
            }
        }
    });
  }

  @Override
  public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[]
grantResults) {
    if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
      if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        // Permission granted, so request location updates.
        requestLocationUpdates();
      } else {
        locationTextView.setText("Location permission denied");
      }
    }
  }
}
```

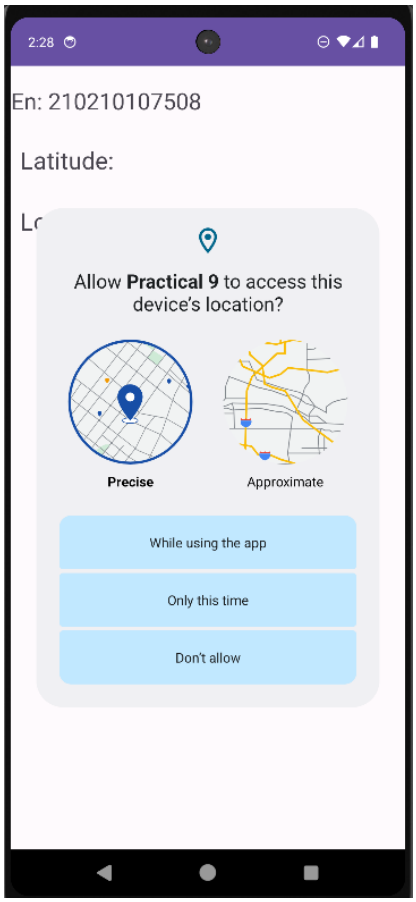
activity_main.xml

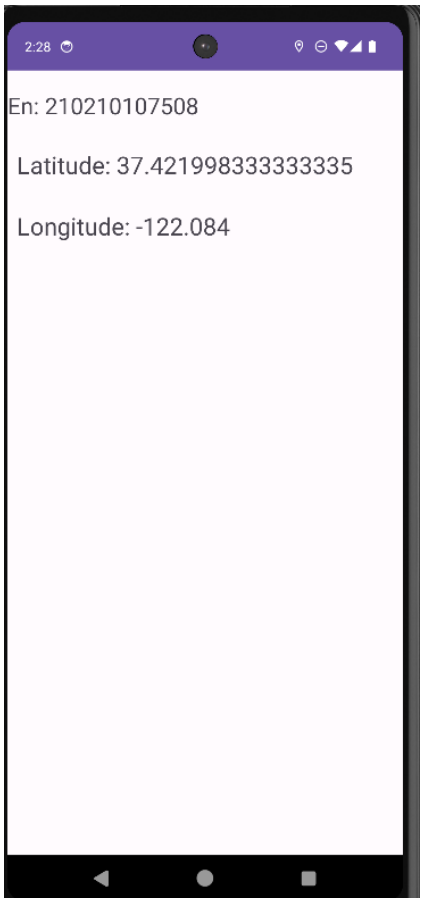
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"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

  <TextView
    android:id="@+id/locationTextView"
    .. />

  <TextView
    android:id="@+id/textView"
    android:layout_width="268dp"
    android:layout_height="61dp"
    android:text="En: 210210107508"
    android:textSize="24sp"
    ..

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

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

Output Screenshots:





References used by the students:

https://developer.android.com/docs
https://www.geeksforgeeks.org/

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						

Practical – 10

Aim: Display the use of animations.

(Write a sample application to display working of android animation)

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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

    final ImageView imageView = findViewById(R.id.imageView);
    Button animateButton = findViewById(R.id.animateButton);

    final Animation fadeInAnimation = AnimationUtils.loadAnimation(this, R.anim.fade_in);

    fadeInAnimation.setAnimationListener(new Animation.AnimationListener() {
      @Override
      public void onAnimationStart(Animation animation) {}

      @Override
      public void onAnimationEnd(Animation animation) {
      }

      @Override
      public void onAnimationRepeat(Animation animation) {}
    });

    animateButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        imageView.startAnimation(fadeInAnimation);
      }
    });

    Button animateButton2 = findViewById(R.id.animateButton2);
    final Animation rotateAnimation = AnimationUtils.loadAnimation(this, R.anim.rotate);

    rotateAnimation.setAnimationListener(new Animation.AnimationListener() {
      @Override
      public void onAnimationStart(Animation animation) {}

      @Override
```

```
            public void onAnimationEnd(Animation animation) {
            }

            @Override
            public void onAnimationRepeat(Animation animation) {}
        });


        animateButton2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                imageView.startAnimation(rotateAnimation);
            }
        });
    }
}
```

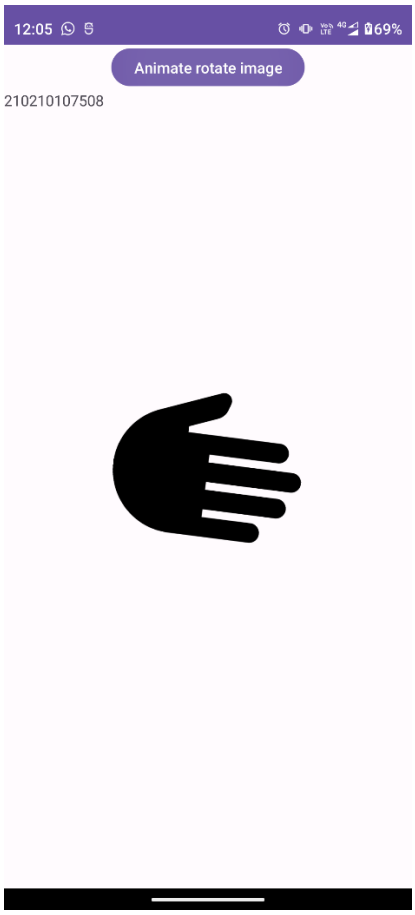
activity_main.xml

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

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:src="@drawable/ic_hand"
         android:layout_centerInParent="true" />

    <Button
        ...
        android:id="@+id/animateButton"  />
    <Button
       ...
        android:id="@+id/animateButton2" />

    <TextView
        android:id="@+id/textView"
        ...
        android:text="210210107508" />
</RelativeLayout>
```

Res/anim/rotate.xml

```
<rotate
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromDegrees="0"
    android:toDegrees="360"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="1000" />
```

Output Screenshots:



References used by the students:

https://developer.android.com/docs
https://stackoverflow.com/
https://www.geeksforgeeks.org/

Rubric wise marks obtained:

RUBRICS	1	2	3	4	5	TOTAL
MARKS						