

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING



MOBILE APPLICATION DEVELOPMENT LABORATORY

Subject Code :KG23ACS322

Regulation : KGR23

Academic Year :2025-2026

III B. TECH II SEMESTER

COMPUTER SCIENCE AND ENGINEERING
KGREDDY COLLEGE OF ENGINEERING AND TECHNOLOGY
Affiliated by JNTUH, Chilkur, (V), Moinabad (M) R.R Dist, TS-501504

VISION AND MISSION OF THE INSTITUTION

VISION:

To become an institution which is internationally recognized for its holistic approach to engineering, innovative teaching and learning culture, research and entrepreneurial ecosystem, And sustainable social impact in the community.

MISSION:

- To offer undergraduate and post-graduate programs which are supported through industry relevant curriculum and innovative teaching and learning processes that would help students succeed in their professional careers.
- To provide faculty and students with an ecosystem that fosters innovation, research, entrepreneurship, and international exposure through strategic partnerships with government organizations and collaboration with industries.
- To provide holistic learning environment to students which will contribute to their personal and Professional growth and enable them to become leaders in the irrespective fields.
- To contribute to the development of the region by using our technological expertise to work with nearby communities and support them in their social and economic development.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION:

To be recognized as a department of excellence by stimulating a learning environment in which students and faculty will thrive and grow to achieve their professional, institutional and societal goals.

MISSION:

- To provide high quality technical education to students that will enable life-long learning and Build expertise in advanced technologies in Computer Science and Engineering.
- To promote research and development by providing opportunities to solve complex engineering problems in collaboration with industry and government agencies.
- To encourage professional development of students that will inculcate ethical values and leadership skills through entrepreneurship while working with the community to address societal issues.

PROGRAM EDUCATIONAL OBJECTIVES

PEO1: Graduates will provide solutions to difficult and challenging issues in their profession by applying computer science and engineering theory and principles.

PEO2: Graduates have successful careers in computer science and engineering fields or will be able to successfully pursue advanced degrees.

PEO3: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism, moral and ethical responsibility.

PEO4: Graduates will develop the ability to understand and analyze engineering issues in a broader perspective with ethical responsibility towards sustainable development.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PROGRAM OUTCOMES

<p>POI: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.</p>
<p>POII: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems, reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.</p>
<p>POIII: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet specified needs with appropriate consideration for public health and safety, and cultural, societal, and environmental factors.</p>
<p>PO IV: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.</p>
<p>POV: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modeling, to complex engineering activities with an understanding of limitations.</p>
<p>POVI: The engineer and society: Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal, and cultural issues and the responsibilities relevant to professional engineering practice.</p>
<p>POVII: Environment and sustainability: Understand the impact of professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.</p>
<p>POVIII: Ethics: Apply ethical principles and commit to professional ethics, responsibilities, and norms of engineering practice.</p>
<p>POIX: Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams and multidisciplinary settings.</p>
<p>POX: Communication: Communicate effectively on complex engineering activities with the engineering community and society at large, including writing reports, preparing design documentation, making presentations, and giving and receiving clear instructions.</p>
<p>POXI: Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these as a member and leader in a team to manage projects in multidisciplinary environments.</p>
<p>PO XII: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the context of technological change</p>

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM SPECIFIC OUTCOMES

PSO1: Computer Science and Engineering graduates will be able to analyze, design, develop, and test intelligent systems using mathematical foundations and management principles, and apply computational solutions to develop secure applications and hardware prototypes.

PSO2: Graduates will be able to analyze contemporary research issues in various areas of Computer Science and Engineering, identify research gaps, and conduct research in specialized or emerging fields.

PSO3: Graduates will develop skills to solve problems in programming concepts, evaluate environmental and social issues with ethical responsibility, and manage projects in multidisciplinary fields, fostering careers, entrepreneurship, and higher studies.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MOBILE APPLICATION DEVELOPMENT LABORATORY

Course Code: KG23ACS322

LTPC

B.Tech. III Year II Sem

0 0 2 1

Course Objectives: The objectives of this course for the students are to:

1. Learn how to develop Applications in an android environment.
2. Learn how to develop user interface applications.
3. Learn how to develop URL related applications.
4. Program mobile applications for the Android operating system that use basic and advanced phone features
5. Deploy applications to the Android marketplace for distribution.

Course Outcomes:

1. **CO 1:** Understand the working of Android OS Practically. (k2)
2. **CO 2:** Develop user interfaces. (K6)
3. **CO 3:** Develop, deploy and maintain the Android Applications.(k6)
4. **CO 4:** Design persistent data storage for android operating systems. (K6)
5. **CO 5:** Develop Android applications for real-world use. (K6)

List of Experiments

S.No	Name ofThe Experiment
1	A) Create an Android application that shows Hello <name> of the user and run it on an emulator. B) Create an application that takes the name from a text box and shows a hello message along with the name entered in the text box when the user clicks the OK button.
2	Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (DatePicker), State (Spinner), and a Submit button. On clicking the submit button, print all the data below the Submit button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.
3	Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a "Back" button. If the screen is rotated to landscape mode (width greater than height), then the screen should show the list on the left fragment and details on the right fragment instead of the second screen with the back button. Use Fragment transactions and Rotation event listeners.
4	Develop an application that uses a menu with 3 options for dialing a number, opening a website, and sending an SMS. On selecting an option, the appropriate action should be invoked using intents.
5	Develop an application that inserts some notifications into the Notification area and whenever a notification is inserted, it should show a toast with details of the notification.
6	Create an application that uses a text file to store usernames and passwords (tab-separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show a dialog with a Login Failed message.
7	Create a user registration application that stores the user details in a database table.
8	Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.
9	Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.
10	Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number, etc.
11	Create an application that saves user information like name, age, gender, etc. in Shared Preferences and retrieves them when the program restarts.
12	Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a TimePicker to set the alarm time

TEXTBOOK:

1. Professional Android 4 Application Development – Reto Meier, Wiley India (Wrox), 2012.
2. Android Application Development for Java Programmers – James C. Sheusi, Cengage, 2013.

REFERENCE BOOKS:

1. Beginning Android 4 Application Development – Wei-Meng Lee, Wiley India (Wrox), 2013.

Experiment-1: **A)** Create an Android application that shows Hello <name> of the user and run it on an emulator.
B) Create an application that takes the name from a text box and shows a hello message along with the name entered in the text box when the user clicks the OK button.

A) Program:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">
    <TextView
        android:id="@+id/tvHello"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello User"
        android:textSize="24sp"/>
</LinearLayout>
```

MainActivity.java

```
package com.example.helloapp;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView tvHello = findViewById(R.id.tvHello);
        tvHello.setText("Hello John"); // Replace John with your name
    }
}
```

Output:

Hello John

B)Program:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">
    <EditText
        android:id="@+id/etName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your name" />
    <Button
        android:id="@+id/btnOk"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="OK"
        android:layout_marginTop="16dp"/>
    <TextView
        android:id="@+id/tvGreeting"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="24sp"
        android:paddingTop="20dp"/>
</LinearLayout>
```

MainActivity.java

```
package com.example.helloapp;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
```

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        EditText etName = findViewById(R.id.etName);  
        Button btnOk = findViewById(R.id.btnOk);  
        TextView tvGreeting = findViewById(R.id.tvGreeting);  
        btnOk.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                String name = etName.getText().toString();  
                tvGreeting.setText("Hello " + name);  
            }  
        });  
    }  
}
```

Output:

Hello Alice

Experiment-2: Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (DatePicker), State (Spinner), and a Submit button. On clicking the submit button, print all the data below the Submit button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.

Program:

A) Linear Layout Version

activity_main.xml

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">
<EditText android:id="@+id/etUserName" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:hint="User Name"/>
<EditText android:id="@+id/etPassword" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:hint="Password"
    android:inputType="textPassword"/>
<EditText android:id="@+id/etAddress" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:hint="Address" android:inputType="textMultiLine"/>
<TextView android:text="Gender"/>
<RadioGroup android:id="@+id/rgGender" android:orientation="horizontal">
    <RadioButton android:id="@+id/rbMale" android:text="Male"/>
    <RadioButton android:id="@+id/rbFemale" android:text="Female"/>
</RadioGroup>
<EditText android:id="@+id/etAge" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:hint="Age" android:inputType="number"/>

<DatePicker android:id="@+id/dpDOB" android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
<Spinner android:id="@+id/spinnerState" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:entries="@array/states"/>
<Button android:id="@+id/btnSubmit" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Submit" android:layout_marginTop="16dp"/>
```

```
<TextView android:id="@+id/tvResult" android:layout_width="match_parent"  
    android:layout_height="wrap_content" android:text="" android:paddingTop="16dp"/>
```

```
</LinearLayout>
```

```
</ScrollView>
```

MainActivity.java

```
package com.example.userform;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.*;  
import androidx.appcompat.app.AppCompatActivity;  
public class MainActivity extends AppCompatActivity {  
    EditText etUserName, etPassword, etAddress, etAge;  
    RadioGroup rgGender;  
    DatePicker dpDOB;  
    Spinner spinnerState;  
    Button btnSubmit;  
    TextView tvResult;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        etUserName = findViewById(R.id.etUserName);  
        etPassword = findViewById(R.id.etPassword);  
        etAddress = findViewById(R.id.etAddress);  
        etAge = findViewById(R.id.etAge);  
        rgGender = findViewById(R.id.rgGender);  
        dpDOB = findViewById(R.id.dpDOB);  
        spinnerState = findViewById(R.id.spinnerState);  
        btnSubmit = findViewById(R.id.btnSubmit);  
        tvResult = findViewById(R.id.tvResult);  
        btnSubmit.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
  
                String userName = etUserName.getText().toString();
```

```
String password = etPassword.getText().toString();
String address = etAddress.getText().toString();
int selectedGenderId = rgGender.getCheckedRadioButtonId();
RadioButton selectedGender = findViewById(selectedGenderId);
String gender = selectedGender.getText().toString();
String age = etAge.getText().toString();
int day = dpDOB.getDayOfMonth();
int month = dpDOB.getMonth() + 1;
int year = dpDOB.getYear();
String dob = day + "/" + month + "/" + year;
String state = spinnerState.getSelectedItem().toString();
String result = "Name: " + userName + "\nPassword: " + password +
    "\nAddress: " + address + "\nGender: " + gender +
    "\nAge: " + age + "\nDOB: " + dob +
    "\nState: " + state;

tvResult.setText(result);
}
});
}
}
```

Output:

Name: John
Password: 12345
Address: 123 Street
Gender: Male
Age: 25
DOB: 24/3/2000
State: Telangana

B) Relative Layout Version

activity_main.xml

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
android:padding="16dp">
<EditText android:id="@+id/etUserName" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:hint="User Name"/>
<EditText android:id="@+id/etPassword" android:layout_below="@id/etUserName"
    android:layout_width="match_parent" android:layout_height="wrap_content"
    android:hint="Password" android:inputType="textPassword"/>
<EditText android:id="@+id/etAddress" android:layout_below="@id/etPassword"
    android:layout_width="match_parent" android:layout_height="wrap_content"
    android:hint="Address" android:inputType="textMultiLine"/>
<TextView android:id="@+id/tvGender" android:layout_below="@id/etAddress"
    android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Gender"/>
<RadioGroup android:id="@+id/rgGender" android:layout_below="@id/tvGender"
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:orientation="horizontal">
    <RadioButton android:id="@+id/rbMale" android:text="Male"/>
    <RadioButton android:id="@+id/rbFemale" android:text="Female"/>
</RadioGroup>
<EditText android:id="@+id/etAge" android:layout_below="@id/rgGender"
    android:layout_width="match_parent" android:layout_height="wrap_content"
    android:hint="Age" android:inputType="number"/>
<DatePicker android:id="@+id/dpDOB" android:layout_below="@id/etAge"
    android:layout_width="wrap_content" android:layout_height="wrap_content"/>
<Spinner android:id="@+id/spinnerState" android:layout_below="@id/dpDOB"
    android:layout_width="match_parent" android:layout_height="wrap_content"
    android:entries="@array/states"/>
<Button android:id="@+id/btnSubmit" android:layout_below="@id/spinnerState"
    android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Submit"
    android:layout_marginTop="16dp"/>
<TextView android:id="@+id/tvResult" android:layout_below="@id/btnSubmit"
    android:layout_width="match_parent" android:layout_height="wrap_content" android:text=""
    android:paddingTop="16dp"/>
</RelativeLayout>
</ScrollView>
```

Output:

Name: John
Password: 12345
Address: 123 Street
Gender: Male
Age: 25
DOB: 24/3/2000
State: Telangana

C) Table Layout Version

activity_main.xml

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:stretchColumns="1" android:padding="16dp">
    <TableRow>
        <TextView android:text="User Name"/>
        <EditText android:id="@+id/etUserName"/>
    </TableRow>
    <TableRow>
        <TextView android:text="Password"/>
        <EditText android:id="@+id/etPassword" android:inputType="textPassword"/>
    </TableRow>
    <TableRow>
        <TextView android:text="Address"/>
        <EditText android:id="@+id/etAddress" android:inputType="textMultiLine"/>
    </TableRow>
    <TableRow>
        <TextView android:text="Gender"/>
        <RadioGroup android:id="@+id/rgGender" android:orientation="horizontal">
            <RadioButton android:id="@+id/rbMale" android:text="Male"/>
            <RadioButton android:id="@+id/rbFemale" android:text="Female"/>
        </RadioGroup>
    </TableRow>
    <TableRow>
        <TextView android:text="Age"/>
        <EditText android:id="@+id/etAddress" android:inputType="number"/>
    </TableRow>
```

```
<TableRow>
  <TextView android:text="DOB"/>
  <DatePicker android:id="@+id/dpDOB"/>
</TableRow>
<TableRow>
  <TextView android:text="State"/>
  <Spinner android:id="@+id/spinnerState" android:entries="@array/states"/>
</TableRow>
<TableRow>
  <Button android:id="@+id/btnSubmit" android:text="Submit"/>
</TableRow>
<TableRow>
  <TextView android:id="@+id/tvResult" android:layout_span="2" android:paddingTop="16dp"/>
</TableRow>
</TableLayout>
```

Output:

Name: John
Password: 12345
Address: 123 Street
Gender: Male
Age: 25
DOB: 24/3/2000
State: Telangana

Experiment-3: Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a "Back" button. If the screen is rotated to landscape mode (width greater than height), then the screen should show the list on the left fragment and details on the right fragment instead of the second screen with the back button. Use Fragment transactions and Rotation event listeners.

Program:

Candidate Model – Candidate.java

```
package com.example.candidatelist;
import java.io.Serializable;
public class Candidate implements Serializable {
    String name;
    String email;
    String phone;
    public Candidate(String name, String email, String phone) {
        this.name = name;
        this.email = email;
        this.phone = phone;
    }
}
```

MainActivity.java

```
package com.example.candidatelist;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentTransaction;
import android.content.res.Configuration;
public class MainActivity extends AppCompatActivity implements ListFragment.OnItemSelectedListener {
    boolean isTwoPane;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Check orientation
        if (getResources().getConfiguration().orientation == Configuration.ORIENTATION_LANDSCAPE) {
            setContentView(R.layout.activity_main_land);
            isTwoPane = true;
            getSupportFragmentManager().beginTransaction()
                .replace(R.id.list_fragment_container, new ListFragment())
                .commit();
        } else {
            setContentView(R.layout.activity_main);
            isTwoPane = false;

            getSupportFragmentManager().beginTransaction()
                .replace(R.id.fragment_container, new ListFragment())
                .commit();
        }
    }
    @Override
    public void onItemSelected(Candidate candidate) {
        DetailFragment detailFragment = DetailFragment.newInstance(candidate);
```

```

if (isTwoPane) {
    // Landscape → update right fragment
    getSupportFragmentManager().beginTransaction()
        .replace(R.id.detail_fragment_container, detailFragment)
        .commit();
} else {
    // Portrait → replace screen and add to back stack
    FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
    transaction.replace(R.id.fragment_container, detailFragment);
    transaction.addToBackStack(null);
    transaction.commit();
}
}
}

```

ListFragment.java

```

package com.example.candidatelist;
import android.content.Context;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import java.util.ArrayList;
public class ListFragment extends Fragment {
    interface OnItemSelectedListener {
        void onItemSelected(Candidate candidate);
    }
    OnItemSelectedListener listener;
    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        listener = (OnItemSelectedListener) context;
    }
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_list, container, false);
        RecyclerView recyclerView = view.findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
        ArrayList<Candidate> candidates = new ArrayList<>();
        candidates.add(new Candidate("John Doe", "john@example.com", "1234567890"));
        candidates.add(new Candidate("Alice Smith", "alice@example.com", "0987654321"));
        candidates.add(new Candidate("Bob Johnson", "bob@example.com", "1122334455"));
        recyclerView.setAdapter(new CandidateAdapter(candidates));
        return view;
    }
}

```

```

class CandidateAdapter extends RecyclerView.Adapter<CandidateAdapter.ViewHolder> {
    ArrayList<Candidate> list;
    CandidateAdapter(ArrayList<Candidate> list) { this.list = list; }
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.list_item, parent, false);
        return new ViewHolder(view);
    }
    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        Candidate candidate = list.get(position);
        holder.tvName.setText(candidate.name);
        holder.itemView.setOnClickListener(v -> listener.onItemSelected(candidate));
    }
    @Override
    public int getItemCount() { return list.size(); }
    class ViewHolder extends RecyclerView.ViewHolder {
        TextView tvName;
        ViewHolder(View itemView) {
            super(itemView);
            tvName = itemView.findViewById(R.id.tvName);
        }
    }
}

```

DetailFragment.java

```

package com.example.candidatelist;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
public class DetailFragment extends Fragment {
    private static final String ARG_CANDIDATE = "candidate";
    private Candidate candidate;
    public static DetailFragment newInstance(Candidate candidate) {
        DetailFragment fragment = new DetailFragment();
        Bundle args = new Bundle();
        args.putSerializable(ARG_CANDIDATE, candidate);
        fragment.setArguments(args);
        return fragment;
    }
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            candidate = (Candidate) getArguments().getSerializable(ARG_CANDIDATE);

```

```

    }
}
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_detail, container, false);
    TextView tvDetail = view.findViewById(R.id.tvDetail);
    if (candidate != null) {
        tvDetail.setText("Name: " + candidate.name +
            "\nEmail: " + candidate.email +
            "\nPhone: " + candidate.phone);
    }
    return view;
}
}

```

Layout XMLs

Portrait – activity_main.xml

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/fragment_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>

```

Landscape – activity_main_land.xml

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <FrameLayout
        android:id="@+id/list_fragment_container"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="match_parent"/>
    <FrameLayout
        android:id="@+id/detail_fragment_container"
        android:layout_width="0dp"
        android:layout_weight="2"
        android:layout_height="match_parent"/>
</LinearLayout>

```

fragment_list.xml

```

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

```

fragment_detail.xml

```

<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tvDetail"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:textSize="18sp"/>

```

list_item.xml

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/tvName"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:padding="16dp"  
    android:textSize="18sp"/>
```

Output:

John Doe
Alice Smith
Bob Johnson

Name: Alice Smith
Email: alice@example.com
Phone: 0987654321

Experiment-4: Develop an application that uses a menu with 3 options for dialing a number, opening a website, and sending an SMS. On selecting an option, the appropriate action should be invoked using intents.

Program:

MainActivity.java

```
package com.example.menuintents;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    // Inflate menu
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main_menu, menu);
        return true;
    }
    // Handle menu click
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.menu_dial:
                Intent dialIntent = new Intent(Intent.ACTION_DIAL);
                dialIntent.setData(Uri.parse("tel:1234567890"));
                startActivity(dialIntent);
                return true;
            case R.id.menu_website:
                Intent webIntent = new Intent(Intent.ACTION_VIEW);
                webIntent.setData(Uri.parse("https://www.example.com"));
                startActivity(webIntent);
                return true;
            case R.id.menu_sms:
                Intent smsIntent = new Intent(Intent.ACTION_VIEW);
                smsIntent.setData(Uri.parse("sms:1234567890"));
                smsIntent.putExtra("sms_body", "Hello! This is a test SMS.");
                startActivity(smsIntent);
                return true;

            default:
                return super.onOptionsItemSelected(item);
        }
    }
}
```

Menu XML – res/menu/main_menu.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/menu_dial"
    android:title="Dial Number"/>
  <item
    android:id="@+id/menu_website"
    android:title="Open Website"/>
  <item
    android:id="@+id/menu_sms"
    android:title="Send SMS"/>
</menu>
```

Layout XML – activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:gravity="center"
  android:padding="16dp">
  <TextView
    android:text="Select Menu from top-right to perform actions"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"/>
</LinearLayout>
```

Output:

Hello! This is a test SMS.

Experiment-5: Develop an application that inserts some notifications into the Notification area and whenever a notification is inserted, it should show a toast with details of the notification.

Program:

MainActivity.java

```
package com.example.notificationapp;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
public class MainActivity extends AppCompatActivity {
    private final String CHANNEL_ID = "my_channel_id";
    private int notificationId = 1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        createNotificationChannel();
        Button btnNotify = findViewById(R.id.btnNotify);
        btnNotify.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String title = "My Notification " + notificationId;
                String text = "This is notification number " + notificationId;

                // Show toast
                Toast.makeText(MainActivity.this, text, Toast.LENGTH_SHORT).show();
                // Build notification
                NotificationCompat.Builder builder = new NotificationCompat.Builder(MainActivity.this,
CHANNEL_ID)
                    .setSmallIcon(android.R.drawable.ic_dialog_info)
                    .setContentTitle(title)
                    .setContentText(text)
                    .setPriority(NotificationCompat.PRIORITY_DEFAULT);
                // Show notification
                NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
                manager.notify(notificationId, builder.build());

                notificationId++;
            }
        });
    }
    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = "MyChannel";
            String description = "Channel for notifications";
```

```
int importance = NotificationManager.IMPORTANCE_DEFAULT;  
NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);  
channel.setDescription(description);
```

```
NotificationManager manager = getSystemService(NotificationManager.class);  
manager.createNotificationChannel(channel);
```

```
}  
}  
}
```

Layout XML – activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:gravity="center"  
    android:padding="16dp">  
    <Button  
        android:id="@+id/btnNotify"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Create Notification"/>  
</LinearLayout>
```

Output:

This is notification number 1
This is notification number 2
This is notification number 3
This is notification number 4

Experiment-6: Create an application that uses a text file to store usernames and passwords (tab-separated fields and one record per line). When the user submits a login name and password through a screen, the details should be verified with the text file data and if they match, show a dialog saying that login is successful. Otherwise, show a dialog with a Login Failed message.

Program:

Text File – users.txt

```
john    1234
alice   pass123
bob     qwerty
```

Layout XML – activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:gravity="center">
    <EditText
        android:id="@+id/etUsername"
        android:hint="Username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/etPassword"
        android:hint="Password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"/>
    <Button
        android:id="@+id/btnLogin"
        android:text="Login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"/>
</LinearLayout>
```

MainActivity.java

```
package com.example.textfilelogin;
import android.app.AlertDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class MainActivity extends AppCompatActivity {
    EditText etUsername, etPassword;
    Button btnLogin;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
etUsername = findViewById(R.id.etUsername);
etPassword = findViewById(R.id.etPassword);
btnLogin = findViewById(R.id.btnLogin);
btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String username = etUsername.getText().toString().trim();
        String password = etPassword.getText().toString().trim();
        boolean success = checkCredentials(username, password);
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
        if (success) {
            builder.setTitle("Login Status")
                .setMessage("Login Successful")
                .setPositiveButton("OK", null)
                .show();
        } else {
            builder.setTitle("Login Status")
                .setMessage("Login Failed")
                .setPositiveButton("OK", null)
                .show();
        }
    }
});
}

private boolean checkCredentials(String username, String password) {
    try {
        BufferedReader reader = new BufferedReader(new InputStreamReader(getAssets().open("users.txt")));
        String line;
        while ((line = reader.readLine()) != null) {
            String[] parts = line.split("\t"); // tab-separated
            if (parts.length == 2) {
                if (parts[0].equals(username) && parts[1].equals(password)) {
                    reader.close();
                    return true;
                }
            }
        }
        reader.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}
}

```

Output:

App Screen

```
-----  
| Username: [_____] |  
| Password: [_____] |  
| [ Login ] |  
-----
```

Scenario 1 – Correct Login

- Enter:
 - Username: alice
 - Password: pass123

```
-----  
| Login Status |  
| Login Successful |  
| [OK] |  
-----
```

Scenario 2 – Incorrect Login

- Enter:
 - Username: bob
 - Password: wrongpass

```
-----  
| Login Status |  
| Login Failed |  
| [OK] |  
-----
```

Experiment-7: Create a user registration application that stores the user details in a database table.

Program:

Database Helper – DBHelper.java

```
package com.example.userregistration;
import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class DBHelper extends SQLiteOpenHelper {
    public static final String DB_NAME = "UserDB.db";
    public static final String TABLE_NAME = "users";
    public DBHelper(Context context) {
        super(context, DB_NAME, null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NAME +
            "(id INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "name TEXT, email TEXT, phone TEXT, password TEXT)";
        db.execSQL(createTable);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
    public boolean insertUser(String name, String email, String phone, String password) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put("name", name);
        cv.put("email", email);
        cv.put("phone", phone);
        cv.put("password", password);
        long result = db.insert(TABLE_NAME, null, cv);
        return result != -1;
    }
}
```

Layout XML – activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:gravity="center">
    <EditText
        android:id="@+id/etName"
        android:hint="Name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
```

```

<EditText
    android:id="@+id/etEmail"
    android:hint="Email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
<EditText
    android:id="@+id/etPhone"
    android:hint="Phone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="phone"/>
<EditText
    android:id="@+id/etPassword"
    android:hint="Password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"/>
<Button
    android:id="@+id/btnRegister"
    android:text="Register"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"/>
</LinearLayout>

```

MainActivity.java

```

package com.example.userregistration;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    EditText etName, etEmail, etPhone, etPassword;
    Button btnRegister;
    DBHelper dbHelper;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etName = findViewById(R.id.etName);
        etEmail = findViewById(R.id.etEmail);
        etPhone = findViewById(R.id.etPhone);
        etPassword = findViewById(R.id.etPassword);
        btnRegister = findViewById(R.id.btnRegister);
        dbHelper = new DBHelper(this);
        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String name = etName.getText().toString().trim();
                String email = etEmail.getText().toString().trim();

```

```
String phone = etPhone.getText().toString().trim();
String password = etPassword.getText().toString().trim();
if(name.isEmpty() || email.isEmpty() || phone.isEmpty() || password.isEmpty()) {
    Toast.makeText(MainActivity.this, "Please fill all fields", Toast.LENGTH_SHORT).show();
} else {
    boolean inserted = dbHelper.insertUser(name, email, phone, password);
    if(inserted) {
        Toast.makeText(MainActivity.this, "Registration Successful", Toast.LENGTH_SHORT).show();
        etName.setText("");
        etEmail.setText("");
        etPhone.setText("");
        etPassword.setText("");
    } else {
        Toast.makeText(MainActivity.this, "Registration Failed", Toast.LENGTH_SHORT).show();
    }
}
});
}
```

Output:

App Screen

```
-----
| Name: [_____] |
| Email: [_____] |
| Phone: [_____] |
| Password: [_____] |
| [ Register ] |
-----
```

Experiment-8: Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.

Program:

Database Helper – DBHelper.java

```
package com.example.dblogin;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class DBHelper extends SQLiteOpenHelper {
    public static final String DB_NAME = "LoginDB.db";
    public static final String TABLE_NAME = "users";
    public DBHelper(Context context) {
        super(context, DB_NAME, null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        // Create table
        String createTable = "CREATE TABLE " + TABLE_NAME +
            "(id INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "username TEXT, password TEXT)";
        db.execSQL(createTable);

        // Insert initial users
        insertUser(db, "john", "1234");
        insertUser(db, "alice", "pass123");
        insertUser(db, "bob", "qwerty");
    }
    private void insertUser(SQLiteDatabase db, String username, String password) {
        ContentValues cv = new ContentValues();
        cv.put("username", username);
        cv.put("password", password);
        db.insert(TABLE_NAME, null, cv);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
    // Verify login
    public boolean checkLogin(String username, String password) {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.query(TABLE_NAME,
            new String[]{"id"},
            "username=? AND password=?",
            new String[]{username, password},
            null, null, null);
    }
}
```

```
        boolean exists = cursor.getCount() > 0;
        cursor.close();
        return exists;
    }
}
```

Layout XML – activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    android:gravity="center">
    <EditText
        android:id="@+id/etUsername"
        android:hint="Username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/etPassword"
        android:hint="Password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"/>
    <Button
        android:id="@+id/btnLogin"
        android:text="Login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"/>
</LinearLayout>
```

MainActivity.java

```
package com.example.dblogin;
import android.app.AlertDialog;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    EditText etUsername, etPassword;
    Button btnLogin;
    DBHelper dbHelper;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        etUsername = findViewById(R.id.etUsername);
        etPassword = findViewById(R.id.etPassword);
        btnLogin = findViewById(R.id.btnLogin);
        dbHelper = new DBHelper(this);
    }
}
```

```

btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String username = etUsername.getText().toString().trim();
        String password = etPassword.getText().toString().trim();
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
        if (dbHelper.checkLogin(username, password)) {
            builder.setTitle("Login Status")
                .setMessage("Login Successful")
                .setPositiveButton("OK", null)
                .show();
        } else {
            builder.setTitle("Login Status")
                .setMessage("Login Failed")
                .setPositiveButton("OK", null)
                .show();
        }
    }
});
}
}

```

Output:

```

-----
| Username: [_____] |
| Password: [_____] |
| [ Login ] |
-----

```

Scenario 1 – Correct Login

- Enter:
 - Username: alice
 - Password: pass123

```

-----
| Login Status |
| Login Successful |
| [OK] |
-----

```

Scenario 2 – Incorrect Login

- Enter:
 - Username: bob
 - Password: wrongpass

```

-----
| Login Status |
| Login Failed |
| [OK] |
-----

```

Experiment-9: Create an admin application for the user table, which shows all records as a list and the admin can select any record for edit or modify. The results should be reflected in the table.

Program:

Database Helper – DBHelper.java

```
package com.example.adminapp;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import java.util.ArrayList;
public class DBHelper extends SQLiteOpenHelper {
    public static final String DB_NAME = "UserDB.db";
    public static final String TABLE_NAME = "users";
    public DBHelper(Context context) {
        super(context, DB_NAME, null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NAME +
            "(id INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "name TEXT, email TEXT, phone TEXT, password TEXT)";
        db.execSQL(createTable);

        // Insert initial users
        insertUser(db, "John", "john@example.com", "1234567890", "1234");
        insertUser(db, "Alice", "alice@example.com", "9876543210", "pass123");
        insertUser(db, "Bob", "bob@example.com", "1122334455", "qwerty");
    }
    private void insertUser(SQLiteDatabase db, String name, String email, String phone, String password) {
        ContentValues cv = new ContentValues();
        cv.put("name", name);
        cv.put("email", email);
        cv.put("phone", phone);
        cv.put("password", password);
        db.insert(TABLE_NAME, null, cv);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
    // Get all users
    public ArrayList<User> getAllUsers() {
        ArrayList<User> list = new ArrayList<>();
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery("SELECT * FROM " + TABLE_NAME, null);
        if (cursor.moveToFirst()) {
            do {
                int id = cursor.getInt(0);
```

```

        String name = cursor.getString(1);
        String email = cursor.getString(2);
        String phone = cursor.getString(3);
        String password = cursor.getString(4);
        list.add(new User(id, name, email, phone, password));
    } while (cursor.moveToNext());
}
cursor.close();
return list;
}
// Update user
public boolean updateUser(User user) {
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put("name", user.name);
    cv.put("email", user.email);
    cv.put("phone", user.phone);
    cv.put("password", user.password);
    int result = db.update(TABLE_NAME, cv, "id=?", new String[]{String.valueOf(user.id)});
    return result > 0;
}
}

```

User Model – User.java

```

package com.example.adminapp;
import java.io.Serializable;
public class User implements Serializable {
    int id;
    String name, email, phone, password;
    public User(int id, String name, String email, String phone, String password) {
        this.id = id;
        this.name = name;
        this.email = email;
        this.phone = phone;
        this.password = password;
    }
    @Override
    public String toString() {
        return name + " - " + email;
    }
}

```

Layout XML – activity_main.xml

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">
    <ListView
        android:id="@+id/lvUsers"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>

```

Layout XML – dialog_edit_user.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/etName"
        android:hint="Name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/etEmail"
        android:hint="Email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/etPhone"
        android:hint="Phone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="phone"/>
    <EditText
        android:id="@+id/etPassword"
        android:hint="Password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"/>
</LinearLayout>
```

MainActivity.java

```
package com.example.adminapp;
import android.app.AlertDialog;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.util.ArrayList;
public class MainActivity extends AppCompatActivity {
    ListView lvUsers;
    DBHelper dbHelper;
    ArrayList<User> userList;
    ArrayAdapter<User> adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
lvUsers = findViewById(R.id.lvUsers);
dbHelper = new DBHelper(this);
loadUsers();
lvUsers.setOnItemClickListener((parent, view, position, id) -> {
    User user = userList.get(position);
    showEditDialog(user);
});
}
private void loadUsers() {
    userList = dbHelper.getAllUsers();
    adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1, userList);
    lvUsers.setAdapter(adapter);
}
private void showEditDialog(User user) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Edit User");
    View view = LayoutInflater.from(this).inflate(R.layout.dialog_edit_user, null);
    EditText etName = view.findViewById(R.id.etName);
    EditText etEmail = view.findViewById(R.id.etEmail);
    EditText etPhone = view.findViewById(R.id.etPhone);
    EditText etPassword = view.findViewById(R.id.etPassword);

    // Pre-fill data
    etName.setText(user.name);
    etEmail.setText(user.email);
    etPhone.setText(user.phone);
    etPassword.setText(user.password);
    builder.setView(view);
    builder.setPositiveButton("Update", (dialog, which) -> {
        user.name = etName.getText().toString().trim();
        user.email = etEmail.getText().toString().trim();
        user.phone = etPhone.getText().toString().trim();
        user.password = etPassword.getText().toString().trim();
        boolean updated = dbHelper.updateUser(user);
        if (updated) {
            Toast.makeText(this, "User Updated", Toast.LENGTH_SHORT).show();
            loadUsers(); // refresh list
        } else {
            Toast.makeText(this, "Update Failed", Toast.LENGTH_SHORT).show();
        }
    });
    builder.setNegativeButton("Cancel", null);
    builder.show();
}
}
```

Output:

Initial List

John - john@example.com

Alice - alice@example.com

Bob - bob@example.com

Experiment-10: Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number, etc.

Program:

XML

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.contactsapp">
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <application
        android:allowBackup="true"
        android:label="Contacts App"
        android:supportsRtl="true"
        android:theme="@style/Theme.ContactsApp">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ListView
        android:id="@+id/contactsListView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>

package com.example.contactsapp
import android.Manifest
import android.content.pm.PackageManager
```

```
import android.os.Bundle
import android.provider.ContactsContract
import android.widget.AdapterView
import android.widget.ListView
import android.widget.Toast
import androidx.activity.result.contract.ActivityResultContracts
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat

class MainActivity : AppCompatActivity() {
    private lateinit var contactsListView: ListView
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        contactsListView = findViewById(R.id.contactsListView)
        // Request permission
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.READ_CONTACTS) !=
            PackageManager.PERMISSION_GRANTED) {
            requestPermissionLauncher.launch(Manifest.permission.READ_CONTACTS)
        } else {
            loadContacts()
        }
    }
    private val requestPermissionLauncher = registerForActivityResult(
        ActivityResultContracts.RequestPermission()
    ) { isGranted: Boolean ->
        if (isGranted) {
            loadContacts()
        } else {
            Toast.makeText(this, "Permission denied", Toast.LENGTH_SHORT).show()
        }
    }
    private fun loadContacts() {
        val contactsList = ArrayList<String>()
        val cursor = contentResolver.query(
            ContactsContract.Contacts.CONTENT_URI,
```

```

null,
null,
null,
ContactsContract.Contacts.DISPLAY_NAME + " ASC"
)
if (cursor != null && cursor.count > 0) {
    while (cursor.moveToNext()) {
        val id = cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts._ID))
        val name = cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME))

        val phoneCursor = contentResolver.query(
            ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
            null,
            "${ContactsContract.CommonDataKinds.Phone.CONTACT_ID} = ?",
            arrayOf(id),
            null
        )
        if (phoneCursor != null) {
            while (phoneCursor.moveToNext()) {
                val phoneNumber = phoneCursor.getString(
                    phoneCursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER)
                )
                val type = phoneCursor.getInt(
                    phoneCursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.TYPE)
                )
                val typeLabel = ContactsContract.CommonDataKinds.Phone.getTypeLabel(resources, type,
""").toString()
                contactsList.add("$name - $typeLabel: $phoneNumber")
            }
            phoneCursor.close()
        }
    }
}
cursor.close()
}

```

```
val adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1, contactsList)
```

```
contactsListView.adapter = adapter
```

```
}
```

```
}
```

Output:

Name	Phone Type	Number
-----	-----	-----
Alice	Mobile	987-654-3210
Bob	Home	234-567-8901
Charlie	Mobile	765-432-1098

Experiment-11: Create an application that saves user information like name, age, gender, etc. in Shared Preferences and retrieves them when the program restarts.

Program:

XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_height="match_parent">
    <EditText
        android:id="@+id/nameEditText"
        android:hint="Name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/ageEditText"
        android:hint="Age"
        android:inputType="number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/genderEditText"
        android:hint="Gender"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <Button
        android:id="@+id/saveButton"
        android:text="Save"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <TextView
        android:id="@+id/displayTextView"
        android:text="Saved Info will appear here"
        android:paddingTop="16dp"
```

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

package com.example.sharedprefsapp
import android.content.Context
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
class MainActivity : AppCompatActivity() {
    private lateinit var nameEditText: EditText
    private lateinit var ageEditText: EditText
    private lateinit var genderEditText: EditText
    private lateinit var saveButton: Button
    private lateinit var displayTextView: TextView
    private val PREFS_NAME = "UserPrefs"
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        // Initialize views
        nameEditText = findViewById(R.id.nameEditText)
        ageEditText = findViewById(R.id.ageEditText)
        genderEditText = findViewById(R.id.genderEditText)
        saveButton = findViewById(R.id.saveButton)
        displayTextView = findViewById(R.id.displayTextView)
        // Load saved data if available
        loadUserData()
        // Save button click
        saveButton.setOnClickListener {
            saveUserData()
        }
    }
    private fun saveUserData() {
        val name = nameEditText.text.toString()
    }
}
```

```
val age = ageEditText.text.toString()
val gender = genderEditText.text.toString()
val sharedPref = getSharedPreferences(PREFS_NAME, Context.MODE_PRIVATE)
with(sharedPref.edit()) {
    putString("name", name)
    putString("age", age)
    putString("gender", gender)
    apply()
}
displayTextView.text = "Saved: Name=$name, Age=$age, Gender=$gender"
}
private fun loadUserData() {
    val sharedPref = getSharedPreferences(PREFS_NAME, Context.MODE_PRIVATE)
    val name = sharedPref.getString("name", null)
    val age = sharedPref.getString("age", null)
    val gender = sharedPref.getString("gender", null)
    if (name != null && age != null && gender != null) {
        displayTextView.text = "Saved: Name=$name, Age=$age, Gender=$gender"
    }
}
}
```

Output:

Before Saving:

Saved Info will appear here

After Saving:

Saved: Name=Alice, Age=25, Gender=Female

Experiment-12: Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a TimePicker to set the alarm time

Program:

```
package com.example.sundayalarm
import android.app.AlarmManager
import android.app.PendingIntent
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.media.RingtoneManager
import android.os.Bundle
import android.widget.Button
import android.widget.LinearLayout
import android.widget.TimePicker
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import java.util.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        // Create a simple layout programmatically
        val layout = LinearLayout(this)
        layout.orientation = LinearLayout.VERTICAL
        val timePicker = TimePicker(this)
        timePicker.setIs24HourView(true)
        val button = Button(this)
        button.text = "Set Alarm"
        layout.addView(timePicker)
        layout.addView(button)
        setContentView(layout)
        button.setOnClickListener {
            val hour = timePicker.hour
            val minute = timePicker.minute
            setWeeklyAlarm(hour, minute)
        }
    }
}
```

```

}
private fun setWeeklyAlarm(hour: Int, minute: Int) {
    val calendar = Calendar.getInstance()
    calendar.set(Calendar.DAY_OF_WEEK, Calendar.SUNDAY)
    calendar.set(Calendar.HOUR_OF_DAY, hour)
    calendar.set(Calendar.MINUTE, minute)
    calendar.set(Calendar.SECOND, 0)
    calendar.set(Calendar.MILLISECOND, 0)
    // If time has already passed today, schedule for next week
    if (calendar.timeInMillis <= System.currentTimeMillis()) {
        calendar.add(Calendar.WEEK_OF_YEAR, 1)
    }
    val intent = Intent(this, AlarmReceiver::class.java)
    val pendingIntent = PendingIntent.getBroadcast(
        this,
        0,
        intent,
        PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTABLE
    )
    val alarmManager = getSystemService(Context.ALARM_SERVICE) as AlarmManager
    alarmManager.setRepeating(
        AlarmManager.RTC_WAKEUP,
        calendar.timeInMillis,
        AlarmManager.INTERVAL_DAY * 7,
        pendingIntent
    )
    Toast.makeText(
        this,
        "Alarm set for Sunday at ${String.format("%02d:%02d", hour, minute)}",
        Toast.LENGTH_LONG
    ).show()
}
}
// BroadcastReceiver to play alarm ringtone
class AlarmReceiver : BroadcastReceiver() {

```

```
override fun onReceive(context: Context, intent: Intent) {  
    val alarm = RingtoneManager.getRingtone(  
        context,  
        RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM)  
    )  
    alarm.play()  
    Toast.makeText(context, "Alarm! It's Sunday!", Toast.LENGTH_LONG).show()  
}  
}
```

Output:

Alarm set for Sunday at 08:00

Alarm! It's Sunday!