BATCH AND ROLL NO:
EXPERIMENT NO.: 04
TITLE: Design a mobile application to create registration application which having spinner (subject), radio button (gender), qualification (check box), first insert the value and then show the data in show activity.
DATE OF PERFORMANCE:
DATE OF SUBMISSION:

Title: Design a mobile application to create registration application which having spinner (subject), radio button (gender), qualification (check box), first insert the value and then show the data in show activity.

Requirements:
1 Android studio
2. WordPress Themes and plugins.

Theory:

Introduction

In the realm of mobile application development, creating a robust and user-friendly registration system is a fundamental aspect of building interactive and personalized applications. This lab focuses on the design and implementation of a registration application for a mobile platform. The registration form includes essential input elements such as a Spinner for selecting a subject, Radio Buttons for choosing gender, and Checkboxes for indicating qualifications.

Objective of the Lab: The primary objective of this lab is to guide you through the process of designing and implementing a mobile registration application with dynamic and interactive user interfaces. Specifically, you will learn how to incorporate a Spinner for subject selection, Radio Buttons for gender selection, and Checkboxes for indicating qualifications. The lab also emphasizes the importance of data persistence, as you will insert registration data and display it in a separate "Show" activity.

Components of the Registration Form:

1. **Spinner (Subject):**
   - A Spinner is a dropdown menu that allows users to select a subject from a predefined list.
   - The subject selection provides a dynamic and user-friendly way to capture user preferences.

2. **Radio Buttons (Gender):**
   - Radio Buttons are used to present a set of mutually exclusive options.
   - In this registration form, Radio Buttons are employed to allow users to specify their gender.
   -
3. **Checkboxes (Qualification):**
   - Checkboxes enable users to select multiple options from a list of qualifications.
   - The Checkbox group captures user qualifications effectively.

Data Persistence:

- The lab extends beyond form design to focus on data persistence. Upon registration, user data is stored locally, and a separate "Show" activity will be implemented to display the collected registration information.

Lab Prerequisites:

- Basic understanding of mobile application development concepts.
- Familiarity with the chosen development environment (e.g., Android Studio).
- Prior knowledge of programming languages such as Java (for Android).

Steps:

Step 1: Set Up Your Development Environment

- Ensure that you have Android Studio installed and configured on your machine.

Step 2: Create a New Project

- Open Android Studio and create a new project.
- Choose an appropriate project template, such as "Empty Activity" or "Basic Activity."

Step 3: Design the Registration Form Layout

- Open the XML layout file associated with your main activity (e.g., activity_main.xml).
- Add a Spinner for subjects, Radio Buttons for gender, and Checkboxes for qualifications. Define appropriate IDs for each element.

Step 4: Implement the Java Code

- Open the Java or Kotlin file associated with your main activity (e.g., MainActivity.java).

- Initialize the Spinner, Radio Buttons, and Checkboxes in your code by referencing their IDs.
- Populate the Spinner with subject options dynamically or through a predefined list.

Step 5: Handle User Input

- Implement logic to handle user selections from the Spinner, Radio Buttons, and Checkboxes.
- Capture the selected values and store them in variables or data structures.

Step 6: Create a Data Model

- If not already done, create a data model (e.g., a Java or Kotlin class) to represent the registration information, including subject, gender, and qualifications.

Step 7: Store Data Locally

- Implement a mechanism to store the registration data locally. You can use SharedPreferences, a local database (SQLite), or any other preferred storage method.

Step 8: Implement a "Show" Activity

- Create a new activity for displaying the registration data (e.g., ShowActivity).
- Retrieve the stored registration data and display it in the "Show" activity.

Step 9: Test Your Application

- Run your application on an emulator or a physical device.
- Test the registration form by entering data, selecting options, and submitting the form.
- Verify that the data is stored locally and can be successfully displayed in the "Show" activity.

XML Code:

```
<?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">
 <Spinner
 android:id="@+id/spinner"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:entries="@array/subj"
 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintEnd_toEndOf="parent"
 app:layout_constraintHorizontal_bias="0.498"
 app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toTopOf="parent"
 app:layout_constraintVertical_bias="0.11" />
 <RadioGroup
 android:id="@+id/radiogroup"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintEnd_toEndOf="parent"
 app:layout_constraintHorizontal_bias="0.444"
 app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toTopOf="parent"
 app:layout_constraintVertical_bias="0.261">
 <RadioButton
 android:id="@+id/Male"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Male"></RadioButton>
 <RadioButton
 android:id="@+id/Female"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Female">
 </RadioButton>
app:layout_constraintTop_toBottomOf="@+id/radiogroup"
 app:layout_constraintVertical_bias="0.411" />
 <TextView
 android:id="@+id/textView"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:text="Select one subject"
 android:textColor="#673AB7"
 android:textSize="16sp"
 android:textStyle="bold"
 app:layout_constraintBottom_toTopOf="@+id/spinner"
 app:layout_constraintEnd_toEndOf="parent"
 app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView

android:id="@+id/textView3"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Qualifications"

android:textColor="#673AB7"

android:textSize="16sp"

android:textStyle="bold"

app:layout_constraintBottom_toTopOf="@+id/checkBox"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.5"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/radiogroup" />

<TextView

android:id="@+id/textView4"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Select your Gender"

android:textColor="#673AB7"

android:textColorHighlight="#673AB7"

android:textSize="16sp"

android:textStyle="bold"

app:layout_constraintBottom_toTopOf="@+id/radiogroup"

app:layout_constraintEnd_toEndOf="parent"
```

```
app:layout_constraintHorizontal_bias="0.5"
app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toBottomOf="@+id/spinner"
 app:layout_constraintVertical_bias="0.5" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

2. activity_main2.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=".MainActivity2">
 <TextView
 android:id="@+id/textView2"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:textSize="20sp"
 android:textStyle="bold"
 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintEnd_toEndOf="parent"
 app:layout_constraintHorizontal_bias="0.498"
 app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"

 app:layout_constraintVertical_bias="0.238" />

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

JAVA Code:

1. Main_Activity.java

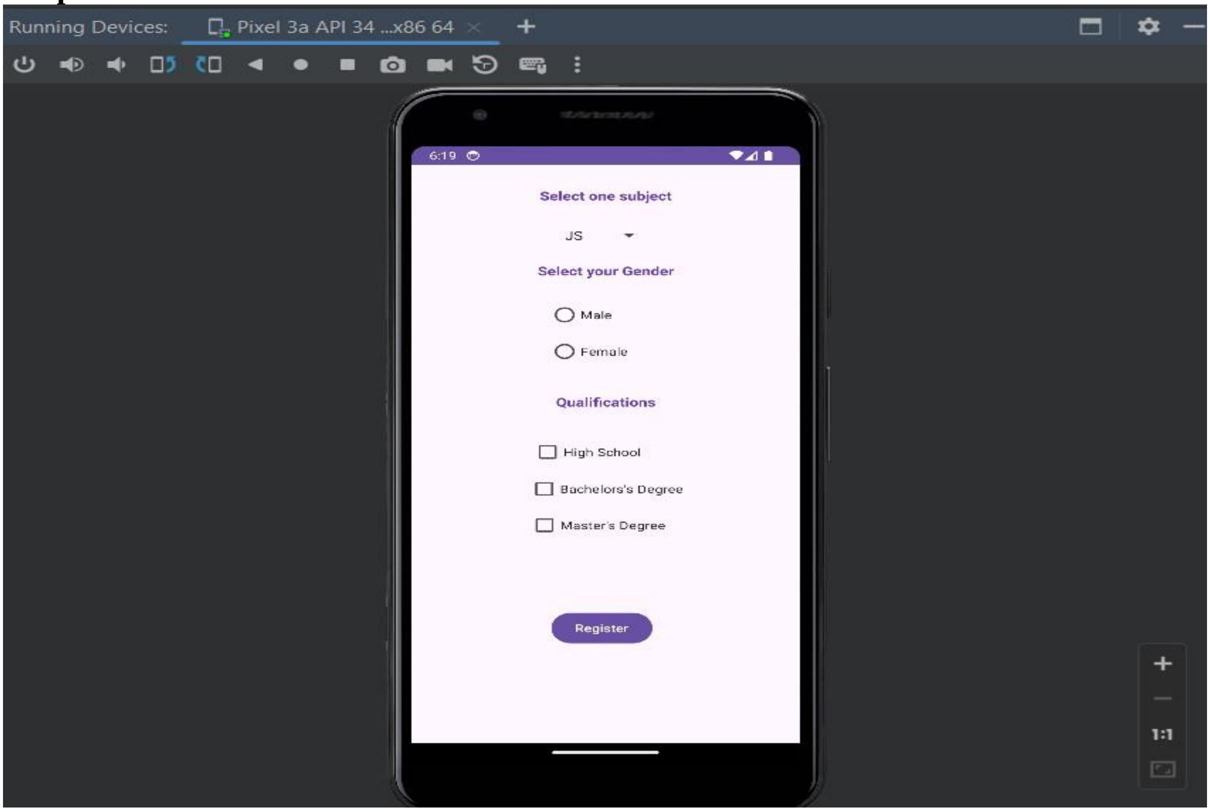
```
package com.example.expt4;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
Spinner spin;
Button btn;
TextView textView;
RadioGroup radioGroup;
CheckBox HighSchool,Bachelor,Master;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);        btn =
findViewById(R.id.button);        spin = findViewById(R.id.spinner);
radioGroup = findViewById(R.id.radiogroup);        HighSchool =
findViewById(R.id.checkBox);
        Bachelor = findViewById(R.id.checkBox2);
        Master = findViewById(R.id.checkBox3);
        Intent intent = new Intent(this, MainActivity2.class);        Bundle extras
= new Bundle();
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String subjects = spin.getSelectedItem().toString();
```

```
              extras.putString("SUB",subjects);            intent.putExtras(extras);
                startActivity(intent);

            }

        });
        Bachelor.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {                boolean checked =
((CheckBox) v).isChecked();                 // Check which checkbox was clicked
                if (checked){                    // Do your coding
                    Toast.makeText(MainActivity.this, "Bachelor's Degree",
Toast.LENGTH_LONG).show();
                }
                else{
                    // Do your coding
                }
            }
        });
        HighSchool.setOnClickListener(new View.OnClickListener() {
@Override
            public void onClick(View v) {
                boolean checked = ((CheckBox) v).isChecked();                // Check
which checkbox was clicked
                if (checked){                    // Do your coding
                    Toast.makeText(MainActivity.this, "High School",
Toast.LENGTH_LONG).show();
                }
                else{
                    // Do your coding
                }
            }
        });
        Master.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                boolean checked = ((CheckBox) v).isChecked();                // Check
which checkbox was clicked
                if (checked){                    // Do your coding
                    Toast.makeText(MainActivity.this, "Master's Degree",
Toast.LENGTH_LONG).show();
                }
                else{
                    // Do your coding
                }
            }
        });

        radioGroup.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                int selid = radioGroup.getCheckedRadioButtonId();
RadioButton radioButton = findViewById(selid);                 String gender =
radioButton.getText().toString();                 extras.putString("Gender",gender);
                intent.putExtras(extras);



            }
        });
    }
```

2. Main_Activity2.java

```
<?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=".MainActivity2">

    <TextView         android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
        android:textSize="20sp"         android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.498"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.238" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:



Select one subject

JS

Select your Gender

○ Male

○ Female

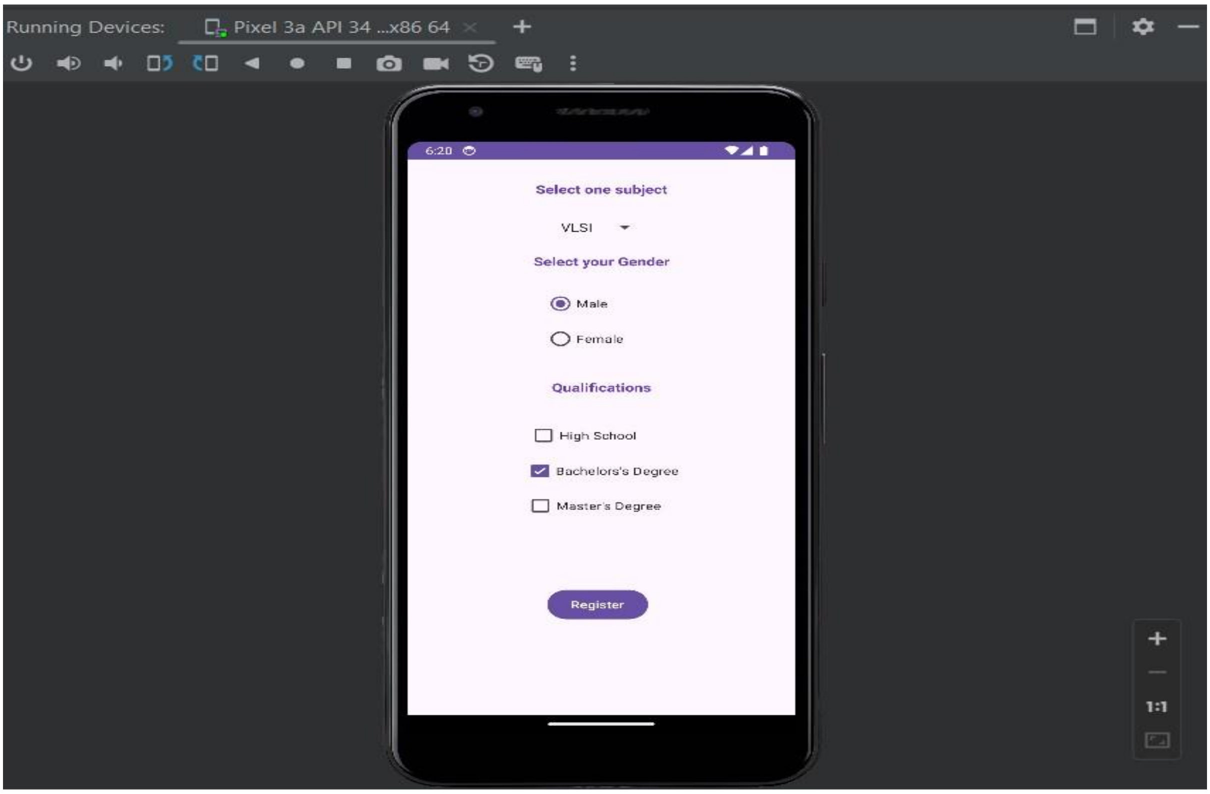
Qualifications

□ High School

□ Bachelors's Degree

□ Master's Degree

Register



Select one subject

VLSI

Select your Gender

● Male

○ Female

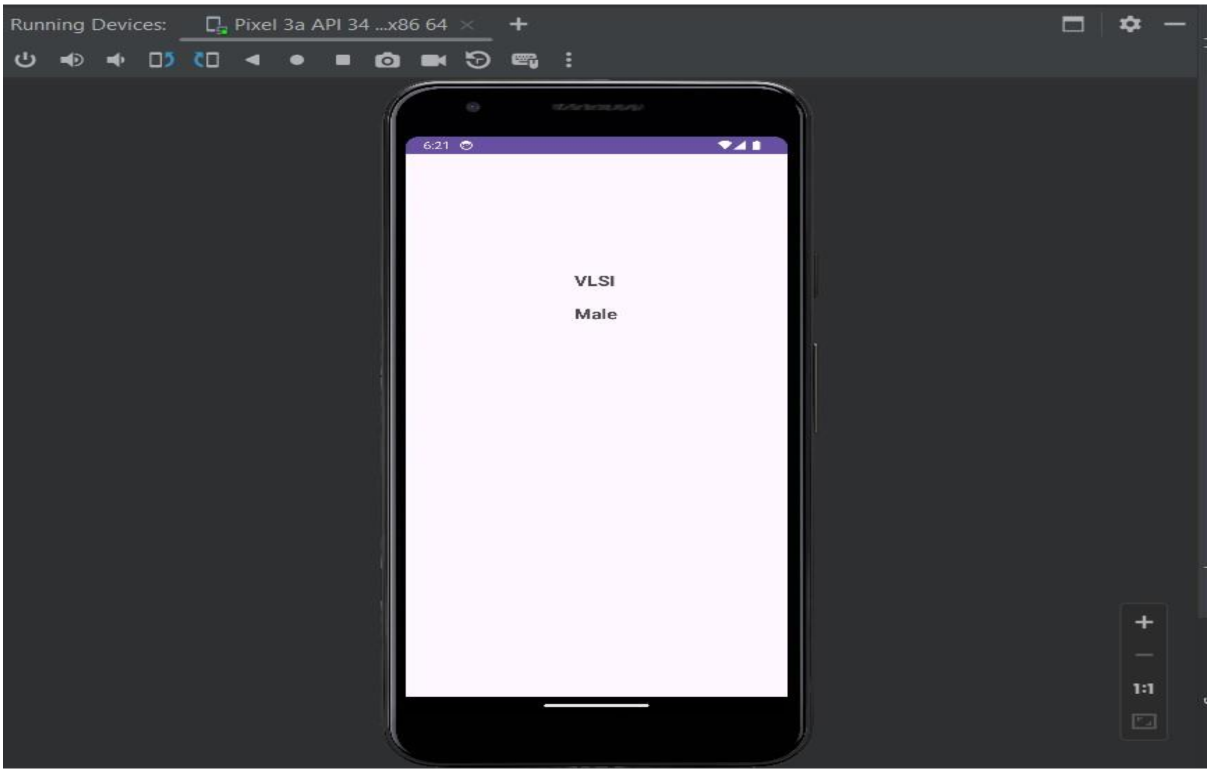
Qualifications

□ High School

☑ Bachelors's Degree

□ Master's Degree

Register



VLSI

Male

Conclusion:

....

....

....