Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 5**	
Title of Lab Assignment: To implement file I/O and Shared Preferences.		
DOP: 04-10-2024	**DOS: 05-10-2024**	
CO Mapped: **CO2, CO3**	**PO Mapped:** **PO2, PO3, PO5, PSO1, PSO2**	**Signature:**

Practical No. 5

Aim: To implement file I/O and Shared Preferences.

1. Program to create a file in a directory and perform following file operations, Write into a file, Read from a file, Delete a file
2. Create a new project and create a login Activity. In this create a login UI asking user email and password with an option of remember me checkbox. Also a button displaying Sign In or Register using shared preferences.

Theory:

In Android, Input/Output (I/O) refers to the operations that involve reading from or writing to external sources such as files, databases, or networks. There are several ways to handle I/O operations in Android:

1. File I/O
    a. Internal Storage: Files are stored in the device's internal storage, and they are private to the application by default.
    b. External Storage: Files can be stored in external storage (SD cards, etc.), and they may be accessible by other apps depending on permissions.

    To perform file operations:
    1. openFileInput() and openFileOutput() for internal storage.
    2. getExternalStorageDirectory() for external storage.
2. Networking I/O:
    a. This involves fetching data from the internet or local networks.
    b. Android provides libraries like HttpURLConnection, OkHttp, and Retrofit for performing network operations.
3. Database I/O:
    a. Android supports local databases using SQLite and Room.
    b. Operations include reading/writing to the database using SQL queries.
4. Content Providers:
    a. Content providers enable apps to share data with other apps.
    b. It can be used for accessing data from contacts, media, etc.

Shared Preferences in Android:

SharedPreferences is a lightweight mechanism used to store small amounts of key-value data that persists across user sessions. It is mainly used to store simple data like user settings, preferences, or app states (e.g., login status).

Key points about Shared Preferences:

1. Stored as an XML file in the app's private internal storage.
2. Data persists even when the app is closed or the device is restarted.
3. It is not designed for large datasets or complex data structures.

SharedPreferences is ideal for small sets of primitive data types such as int, boolean, String, etc.

Code & Output:

1. **Read, Write, Delete**

   MainActivity.java

```
package com.example.practical5a;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {
    private EditText editTextFileName;
    private EditText editTextFileContent;
    private Button buttonWrite;
    private Button buttonRead;
    private Button buttonDelete;

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

```
        buttonWrite = findViewById(R.id.buttonWrite);
        buttonRead = findViewById(R.id.buttonRead);
        buttonDelete = findViewById(R.id.buttonDelete);

        buttonWrite.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String fileName = editTextFileName.getText().toString();
                String fileContent = editTextFileContent.getText().toString();
                writeToFile(fileName, fileContent);
            }
        });

        buttonRead.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String fileName = editTextFileName.getText().toString();
                readFromFile(fileName);
            }
        });

        buttonDelete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String fileName = editTextFileName.getText().toString();
                deleteCustomFile(fileName);
            }
        });
    }

    private void writeToFile(String fileName, String fileContent) {
        File documentsDirectory = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DO
CUMENTS), "");

        if (!documentsDirectory.exists()) {
```

```
            documentsDirectory.mkdirs();
        }

        File file = new File(documentsDirectory, fileName);

        try {
            FileOutputStream fos = new FileOutputStream(file);
            fos.write(fileContent.getBytes());
            fos.close();
            Toast.makeText(this, "File saved at: " + file.getAbsolutePath(),
Toast.LENGTH_SHORT).show();
        } catch (IOException e) {
            e.printStackTrace();
            Toast.makeText(this, "Error saving file", Toast.LENGTH_SHORT).show();
        }
    }

    private void readFromFile(String fileName) {
        File file = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DO
CUMENTS), fileName);
        if (file.exists()) {
            try {
                FileInputStream fis = new FileInputStream(file);
                int c;
                StringBuilder fileContent = new StringBuilder();
                while ((c = fis.read()) != -1) {
                    fileContent.append((char) c);
                }
                fis.close();
                Toast.makeText(this, "File Content: \n" + fileContent.toString(),
Toast.LENGTH_LONG).show();
            } catch (IOException e) {
                e.printStackTrace();
                Toast.makeText(this, "Error reading file", Toast.LENGTH_SHORT).show();
            }
```

```
        } else {
            Toast.makeText(this, "File not found", Toast.LENGTH_SHORT).show();
        }
    }

    private void deleteCustomFile(String fileName) {
        File file = new
File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DO
CUMENTS), fileName);
        boolean deleted = file.delete();
        if (deleted) {
            Toast.makeText(this, "File deleted", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(this, "File not found", Toast.LENGTH_SHORT).show();
        }
    }
}
```

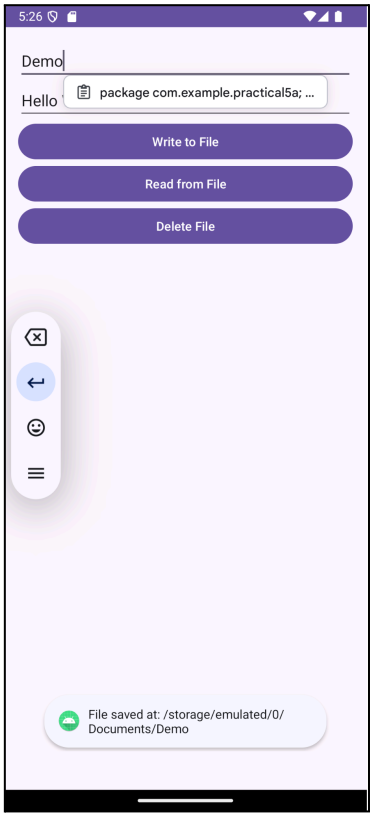
Main_Activity.xml

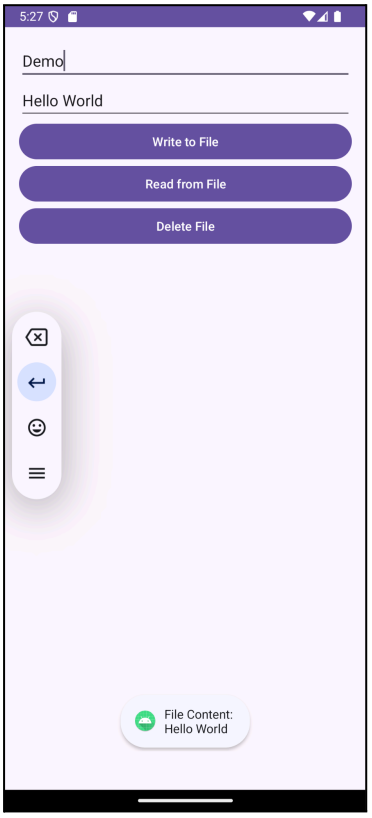
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

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

    <EditText
        android:id="@+id/editTextFileContent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

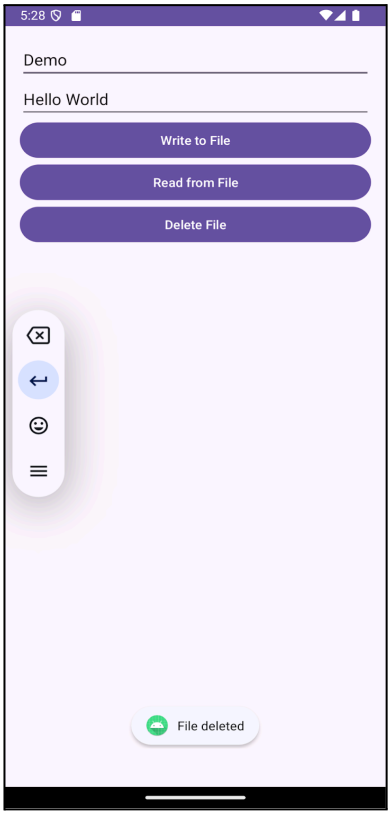
```
        android:hint="File Content" />
    <Button
        android:id="@+id/buttonWrite"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Write to File" />

    <Button
        android:id="@+id/buttonRead"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Read from File" />

    <Button
        android:id="@+id/buttonDelete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Delete File" />
</LinearLayout>
```

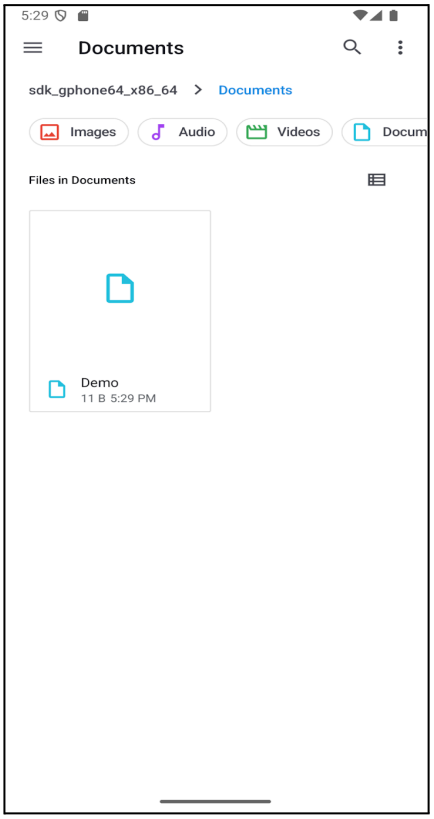
AndroidManifest.xml

```
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```


5:26
Demo
package com.example.practical5a; ...
Hello
Write to File
Read from File
Delete File
File saved at: /storage/emulated/0/
Documents/Demo


5:27
Demo
Hello World
Write to File
Read from File
Delete File
File Content:
Hello World


5:28
Demo
Hello World
Write to File
Read from File
Delete File
File deleted


5:29
Documents
sdk_gphone64_x86_64
Documents
Images
Audio
Videos
Docum
Files in Documents
Demo
11 B 5:29 PM

2. **Login Activity**

MainActivity.java

```
package com.example.practical5b;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private EditText editTextEmail;
    private EditText editTextPassword;
    private CheckBox checkBoxRememberMe;
    private Button buttonSignInOrRegister;
    private SharedPreferences sharedPreferences;

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

        editTextEmail = findViewById(R.id.editTextEmail);
        editTextPassword = findViewById(R.id.editTextPassword);
        checkBoxRememberMe = findViewById(R.id.checkBoxRememberMe);
        buttonSignInOrRegister = findViewById(R.id.buttonSignInOrRegister);

        sharedPreferences = getSharedPreferences("LoginPrefs",
Context.MODE_PRIVATE);
        loadSavedCredentials();

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

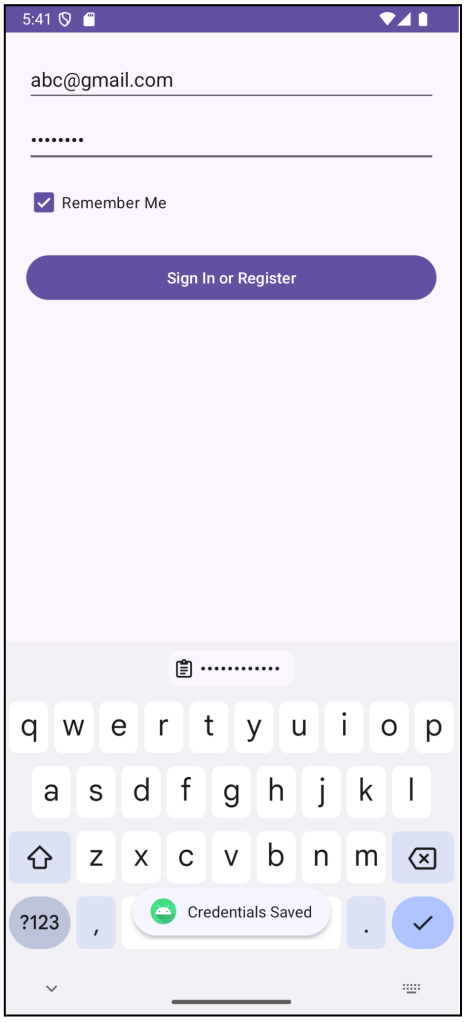
```
            public void onClick(View v) {
                String email = editTextEmail.getText().toString();
                String password = editTextPassword.getText().toString();

                if (email.isEmpty() || password.isEmpty()) {
                    Toast.makeText(LoginActivity.this, "Please enter both email and
password", Toast.LENGTH_SHORT).show();
                } else {
                    if (checkBoxRememberMe.isChecked()) {
                        saveCredentials(email, password);
                    } else {
                        clearSavedCredentials();
                    }
                    Toast.makeText(LoginActivity.this, "Welcome, " + email + "!",
Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    private void saveCredentials(String email, String password) {
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString("Email", email);
        editor.putString("Password", password);
        editor.putBoolean("RememberMe", true);
        editor.apply();
        Toast.makeText(this, "Credentials Saved", Toast.LENGTH_SHORT).show();
    }

    private void loadSavedCredentials() {
        String savedEmail = sharedPreferences.getString("Email", "");
        String savedPassword = sharedPreferences.getString("Password", "");
        boolean rememberMe = sharedPreferences.getBoolean("RememberMe", false);

        if (rememberMe) {
            editTextEmail.setText(savedEmail);
```

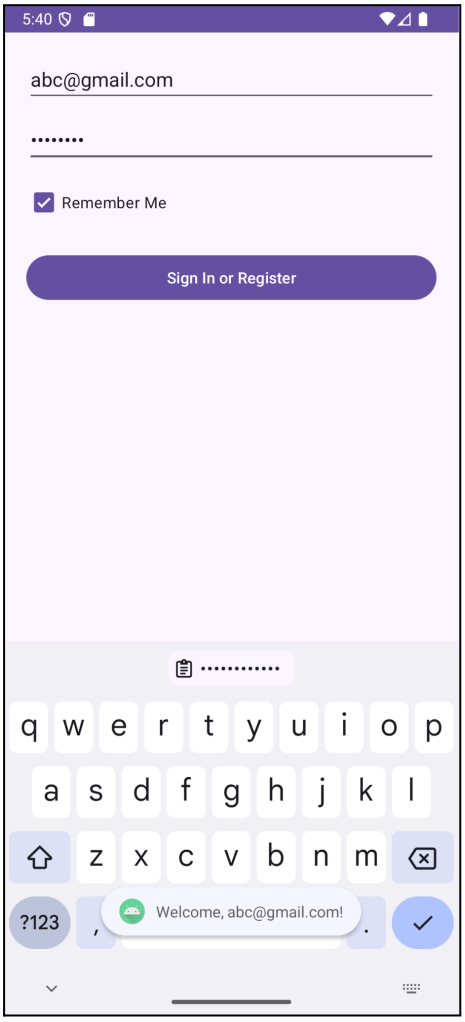
```
            editTextPassword.setText(savedPassword);
            checkBoxRememberMe.setChecked(true);
        }
    }

    private void clearSavedCredentials() {
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.remove("Email");
        editor.remove("Password");
        editor.remove("RememberMe");
        editor.apply();
    }
}
```

Main_Activity.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="20dp">

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Email"
        android:inputType="textEmailAddress"
        android:layout_marginBottom="10dp" />

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Password"
```

```
        android:inputType="textPassword"
        android:layout_marginBottom="10dp" />

    <CheckBox
        android:id="@+id/checkBoxRememberMe"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Remember Me"
        android:layout_marginBottom="20dp" />

    <Button
        android:id="@+id/buttonSignInOrRegister"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Sign In or Register" />

</LinearLayout>
```





Conclusion:

Successfully performed file operations with file I/O and created a simple login system using SharedPreferences for storing user credentials.