Name of Student : Neha Yadav		
Roll Number : 62	**LAB Assignment Number :** 05	
Title of LAB Assignment : 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.		
DOP : 6th September 2024	**DOS :** 15th September 2024	
CO Mapped : CO2, CO3	**PO Mapped :** PO2, PO3, PO5, PSO1, PSO2	**Signature :**

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:

1. File I/O Operations

Objective: Implement file operations in an Android application. You need to create a file, write data to it, read data from it, and delete it.

a. Creating a File

- **Context**: In Android, each application has its own internal storage directory where it can create and manage files. Files created in this directory are private to the application.
- **Process**:
    - **File Creation**: Use the app's internal storage directory to create a new file. This directory is accessible using methods provided by the Android framework.
    - **Naming**: Files are usually named with a `.txt` extension for text files, but you can use other extensions if needed.

b. Writing to a File

- **Context**: Writing data to a file involves opening a file in write mode and sending data to it.
- **Process**:
    - **Open File**: Open the file using the appropriate method (e.g., `FileOutputStream` in Android).
    - **Write Data**: Write data to the file. This can be plain text or binary data.
    - **Close File**: After writing, close the file to ensure all data is properly saved and resources are released.

c. Reading from a File

- **Context**: Reading data involves opening an existing file and retrieving its contents.
- **Process**:
    - **Open File**: Open the file using a method suitable for reading (e.g., `FileInputStream` in Android).
    - **Read Data**: Read data from the file, which may involve processing it line by line or in chunks.
    - **Close File**: After reading, close the file to free up resources.

d. Deleting a File

- **Context**: Deleting a file removes it from the file system.
- **Process**:
    - **Delete File**: Use a method to delete the file from the internal storage. The file should be identified by its name or path.

2. Shared Preferences for Login

Objective: Create a login interface and use shared preferences to save user credentials and login state.

a. Creating a Login UI

- **UI Components**:
    - **Email Field**: An input field where the user can enter their email address.
    - **Password Field**: An input field for the user's password.
    - **Remember Me Checkbox**: An option to remember user credentials for future logins.
    - **Sign In/Register Button**: A button to submit the login or registration information.

b. Using Shared Preferences

- **Context**: Shared Preferences is a way to store simple key-value pairs persistently in Android. It is often used for saving user preferences and settings.
- **Process**:
    - **Save Preferences**: When the user logs in and checks the "Remember Me" option, save their email and login state using Shared Preferences. This involves creating or updating a `SharedPreferences` object and using its editor to put key-value pairs.
    - **Retrieve Preferences**: When the app starts or when needed, retrieve the saved preferences to auto-fill the email field and set the "Remember Me" checkbox based on the stored values.
    - **Clear Preferences**: If the user logs out or the "Remember Me" option is unchecked, clear the stored preferences to ensure credentials are not saved.

CODE:

MainActivity.java

```
package com.example.a40b_practical05;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

   private Button btnFileOperations, btnSharedPreferences;

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

       btnFileOperations = findViewById(R.id.btnFileOperations);
//        btnSharedPreferences = findViewById(R.id.btnSharedPreferences);

       btnFileOperations.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               Intent intent = new Intent(MainActivity.this, FileOperationsActivity.class);
               startActivity(intent);
           }
       });
//
//        btnSharedPreferences.setOnClickListener(new View.OnClickListener() {
//            @Override
//            public void onClick(View v) {
//                Intent intent = new Intent(MainActivity.this, SharedPreferencesActivity.class);
//                startActivity(intent);
//            }
//        });
   }
}
```

activity_main.xml

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

  <Button
     android:id="@+id/btnFileOperations"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:text="File Operations"
     android:layout_centerHorizontal="true"
     android:layout_centerVertical="true"
     android:padding="16dp"/>


</RelativeLayout>
```

FileOperationsActivity.java

```
package com.example.a40b_practical05;

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;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class FileOperationsActivity extends AppCompatActivity {

  private EditText etFileName, etFileContent;
  private Button btnWrite, btnRead, btnDelete;
  private TextView tvFileContent;

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

```
        etFileName = findViewById(R.id.etFileName);
        etFileContent = findViewById(R.id.etFileContent);
        btnWrite = findViewById(R.id.btnWrite);
        btnRead = findViewById(R.id.btnRead);
        btnDelete = findViewById(R.id.btnDelete);
        tvFileContent = findViewById(R.id.tvFileContent);

        btnWrite.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String fileName = etFileName.getText().toString();
                String content = etFileContent.getText().toString();
                writeToFile(fileName, content);
            }
        });

        btnRead.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String fileName = etFileName.getText().toString();
                String content = readFromFile(fileName);
                tvFileContent.setText(content);
            }
        });

        btnDelete.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String fileName = etFileName.getText().toString();
                deleteInternalFile(fileName);
            }
        });
    }

    private void writeToFile(String fileName, String content) {
        File file = new File(getFilesDir(), fileName);
        try (FileOutputStream fos = new FileOutputStream(file)) {
            fos.write(content.getBytes());
            tvFileContent.setText("File written successfully");
        } catch (IOException e) {
            tvFileContent.setText("Error writing file: " + e.getMessage());
        }
    }
```

```
  private String readFromFile(String fileName) {
      File file = new File(getFilesDir(), fileName);
      StringBuilder sb = new StringBuilder();
      try (FileInputStream fis = new FileInputStream(file)) {
          int character;
          while ((character = fis.read()) != -1) {
              sb.append((char) character);
          }
      } catch (IOException e) {
          return "Error reading file: " + e.getMessage();
      }
      return sb.toString();
  }

  private void deleteInternalFile(String fileName) {
      File file = new File(getFilesDir(), fileName);
      if (file.exists()) {
          if (file.delete()) {
              tvFileContent.setText("File deleted successfully");
          } else {
              tvFileContent.setText("Error deleting file");
          }
      } else {
          tvFileContent.setText("File does not exist");
      }
  }
}
```

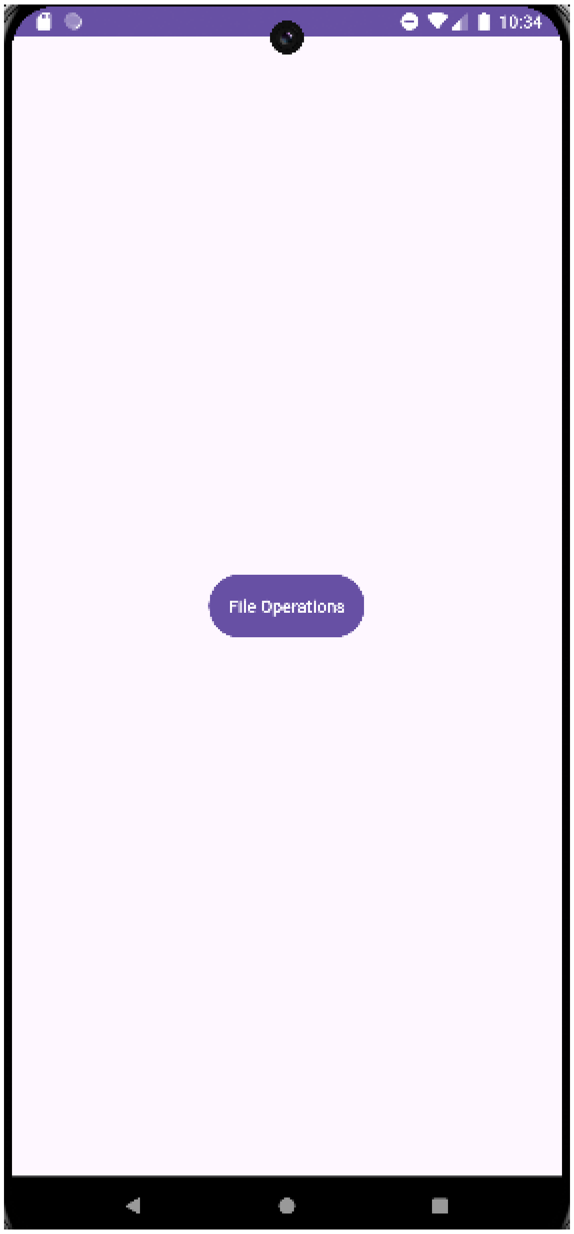
activity_file_operations.xml

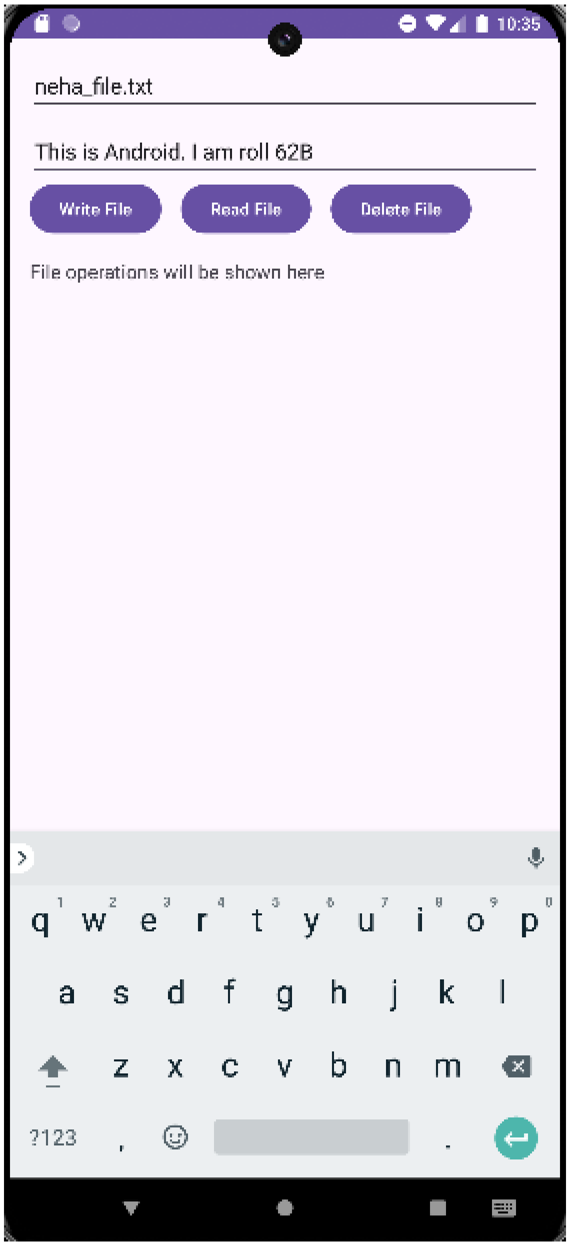
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:padding="16dp">

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

  <EditText
      android:id="@+id/etFileContent"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/etFileName"
        android:hint="File Content"
        android:layout_marginTop="8dp" />

    <Button
        android:id="@+id/btnWrite"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/etFileContent"
        android:text="Write File" />

    <Button
        android:id="@+id/btnRead"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/btnWrite"
        android:layout_marginStart="16dp"
        android:layout_below="@id/etFileContent"
        android:text="Read File" />

    <Button
        android:id="@+id/btnDelete"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/btnRead"
        android:layout_marginStart="16dp"
        android:layout_below="@id/etFileContent"
        android:text="Delete File" />

    <TextView
        android:id="@+id/tvFileContent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/btnWrite"
        android:layout_marginTop="16dp"
        android:text="File operations will be shown here"
        android:textSize="16sp" />

</RelativeLayout>
```

AndroidManifest.xml

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

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

  <application
      android:allowBackup="true"
      android:dataExtractionRules="@xml/data_extraction_rules"
      android:fullBackupContent="@xml/backup_rules"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:supportsRtl="true"
      android:theme="@style/Theme._40B_practical05"
      tools:targetApi="31">
      <activity
          android:name=".SharedPreferencesActivity"
          android:exported="false" />
      <activity
          android:name=".FileOperationsActivity"
          android:exported="false" />
      <activity
          android:name=".MainActivity"
          android:exported="true">
          <intent-filter>
              <action android:name="android.intent.action.MAIN" />

              <category android:name="android.intent.category.LAUNCHER" />
          </intent-filter>
      </activity>
  </application>

</manifest>
```

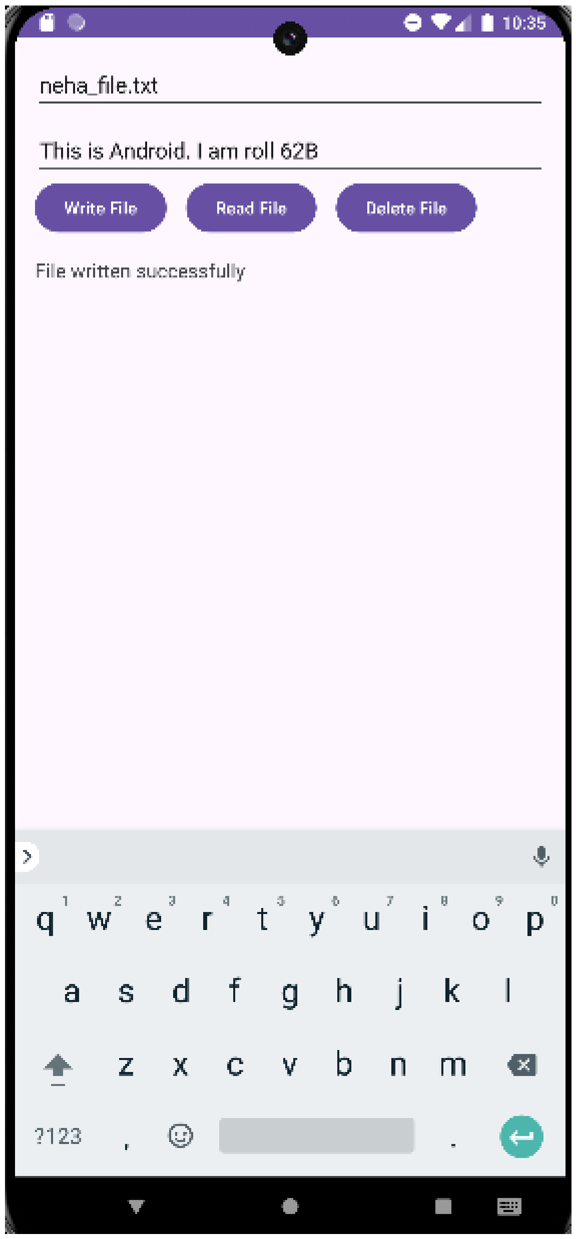




neha_file.txt

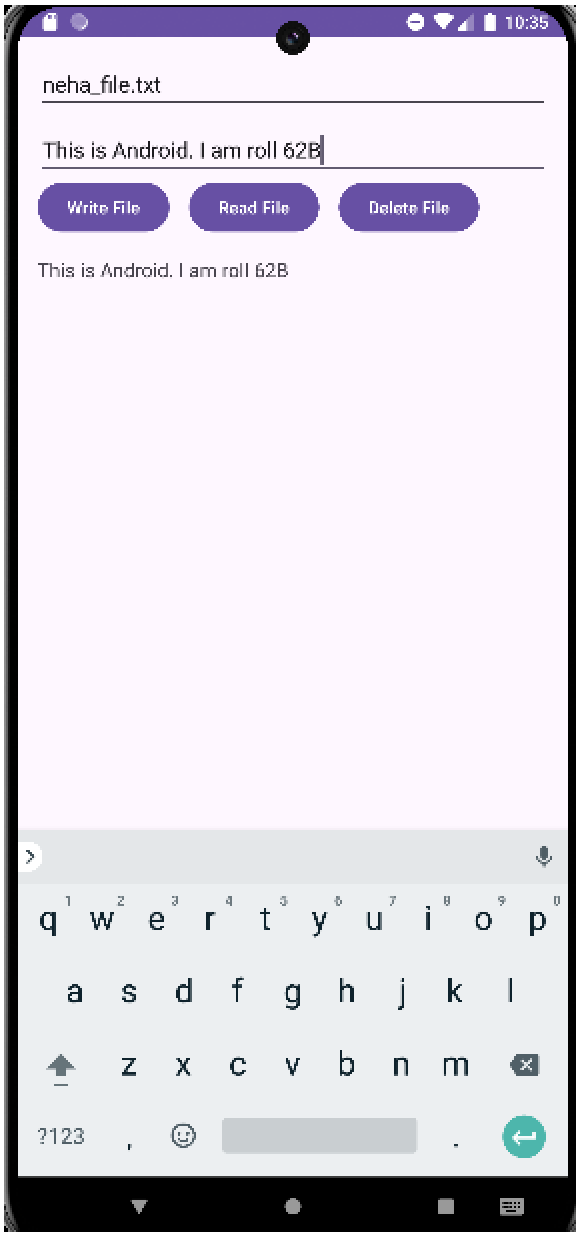
This is Android. I am roll 62B

File operations will be shown here


neha_file.txt

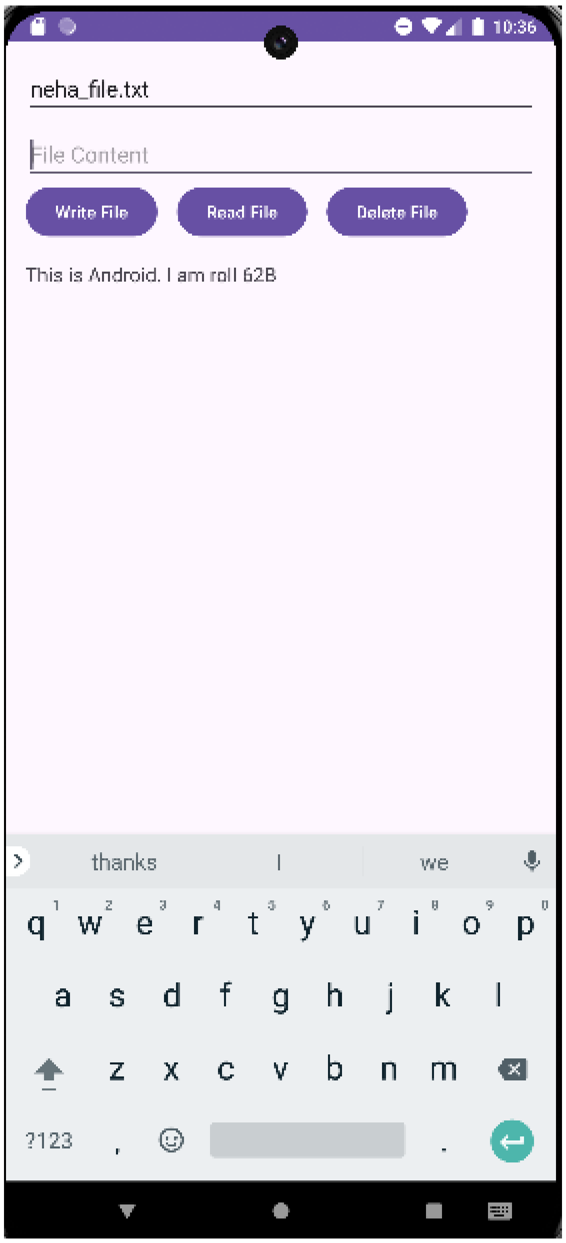
This is Android. I am roll 62B

File written successfully

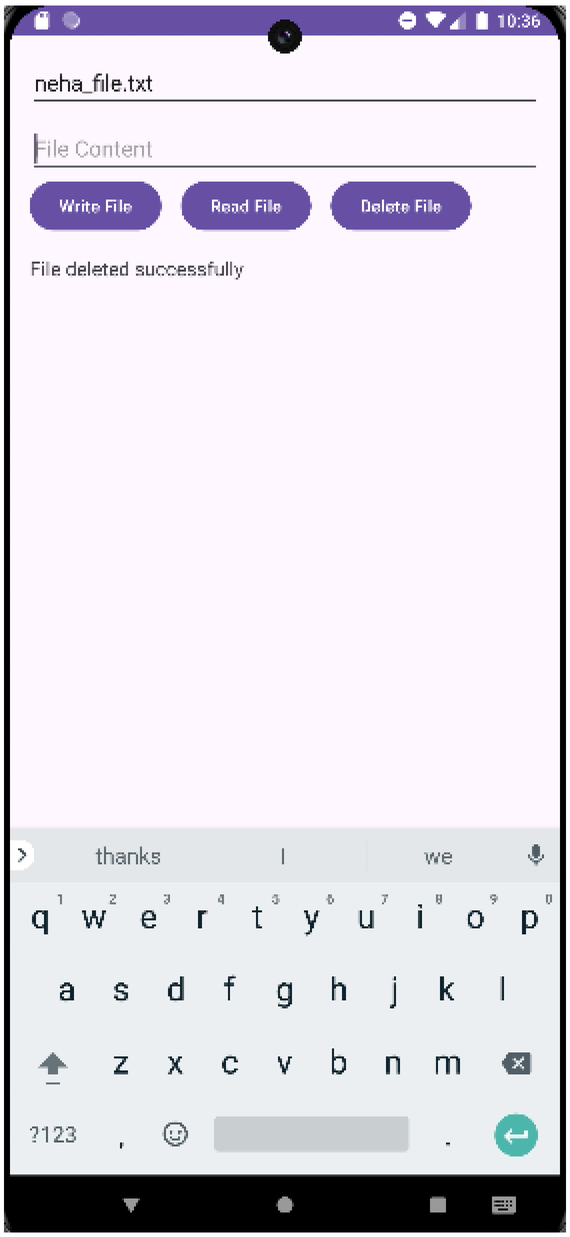

neha_file.txt

This is Android. I am roll 62B

This is Android. I am roll 62B



This is Android. I am roll 62B



File deleted successfully

QUESTION 2

MainActivity.java

```
package com.example.practical05;

import android.content.Context;
import android.content.Intent;
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;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

  private static final String PREFS_NAME = "MyPrefs";
  private static final String PREFS_KEY_EMAIL = "email";
  private static final String PREFS_KEY_PASSWORD = "password";
  private static final String PREFS_KEY_REMEMBER_ME = "rememberMe";

  private EditText emailEditText;
  private EditText passwordEditText;
  private CheckBox rememberMeCheckBox;
  private Button actionButton;

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

     emailEditText = findViewById(R.id.email);
     passwordEditText = findViewById(R.id.password);
     rememberMeCheckBox = findViewById(R.id.rememberMe);
     actionButton = findViewById(R.id.actionButton);

     // Load saved credentials if "Remember Me" was checked
     SharedPreferences prefs = getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE);
     boolean rememberMe = prefs.getBoolean(PREFS_KEY_REMEMBER_ME, false);
     if (rememberMe) {
        emailEditText.setText(prefs.getString(PREFS_KEY_EMAIL, ""));
        passwordEditText.setText(prefs.getString(PREFS_KEY_PASSWORD, ""));
```

```
            rememberMeCheckBox.setChecked(true);
        }

        actionButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                handleLogin();
            }
        });
    }

    private void handleLogin() {
        String email = emailEditText.getText().toString();
        String password = passwordEditText.getText().toString();
        boolean rememberMe = rememberMeCheckBox.isChecked();

        if (email.isEmpty() || password.isEmpty()) {
            Toast.makeText(this, "Please enter email and password",
Toast.LENGTH_SHORT).show();
            return;
        }

        // Example logic: Check email and password validity (mocked here)
        if (isValidCredentials(email, password)) {
            // Save "Remember Me" preference
            SharedPreferences.Editor editor = getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE).edit();
            if (rememberMe) {
                editor.putString(PREFS_KEY_EMAIL, email);
                editor.putString(PREFS_KEY_PASSWORD, password);
            } else {
                editor.remove(PREFS_KEY_EMAIL);
                editor.remove(PREFS_KEY_PASSWORD);
            }
            editor.putBoolean(PREFS_KEY_REMEMBER_ME, rememberMe);
            editor.apply();

            // Navigate to the home screen
            Intent intent = new Intent(MainActivity.this, HomeActivity.class);
            startActivity(intent);
            finish();
        } else {
            Toast.makeText(this, "Invalid credentials", Toast.LENGTH_SHORT).show();
        }
```

```java
    }

    private boolean isValidCredentials(String email, String password) {
        // Mocked method to validate email and password
        return email.equals("neha@gmail.com") && password.equals("password123");
    }
}
```

activity_main.xml

```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/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress" />

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword" />

    <CheckBox
        android:id="@+id/rememberMe"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Remember Me" />

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

</LinearLayout>
```

HomeActivity.java

```java
package com.example.practical05;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class HomeActivity extends AppCompatActivity {

  private static final String PREFS_NAME = "MyPrefs";
  private static final String PREFS_KEY_EMAIL = "email";

  private TextView welcomeText;
  private Button logoutButton;

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

     welcomeText = findViewById(R.id.welcomeText);
     logoutButton = findViewById(R.id.logoutButton);

     // Load the email from Shared Preferences and set it in the welcome message
     SharedPreferences prefs = getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE);
     String email = prefs.getString(PREFS_KEY_EMAIL, "Guest");
     welcomeText.setText("Welcome, " + email);

     logoutButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
           handleLogout();
        }
     });
  }

  private void handleLogout() {
     // Clear the email and password from Shared Preferences
```

```
        SharedPreferences.Editor editor = getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE).edit();
        editor.remove(PREFS_KEY_EMAIL);
        editor.remove("password"); // Note: This line will clear the password from Shared
Preferences
        editor.putBoolean("rememberMe", false);
        editor.apply();

        // Navigate back to the login screen
        Intent intent = new Intent(HomeActivity.this, MainActivity.class);
        startActivity(intent);
        finish();
    }
}
```

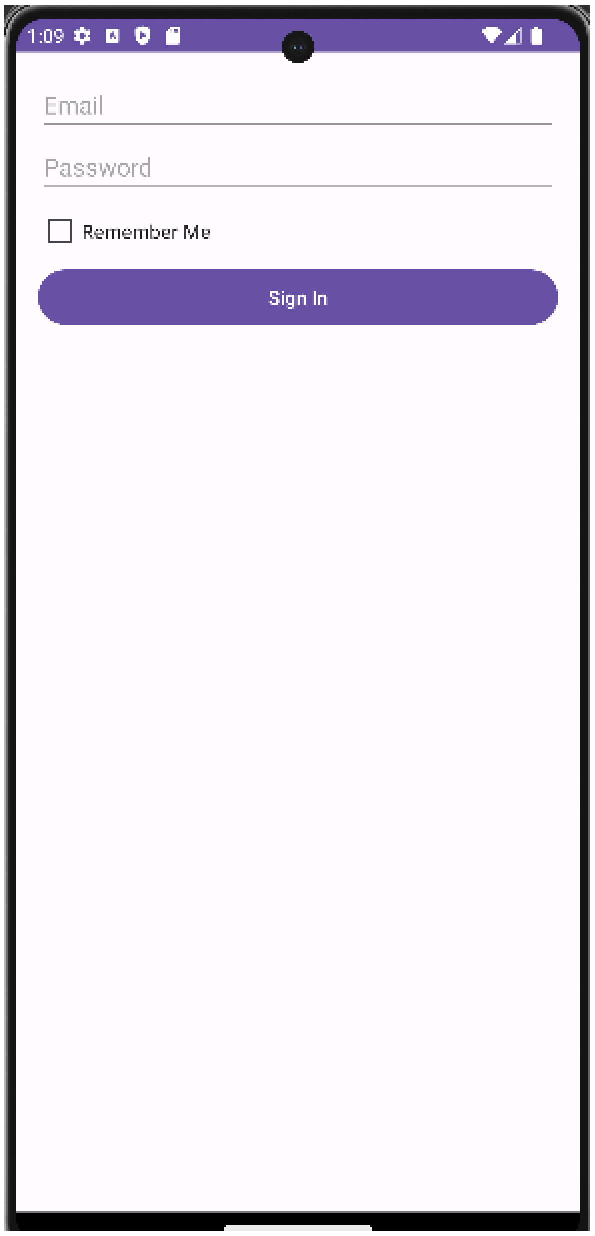
activity_home.xml

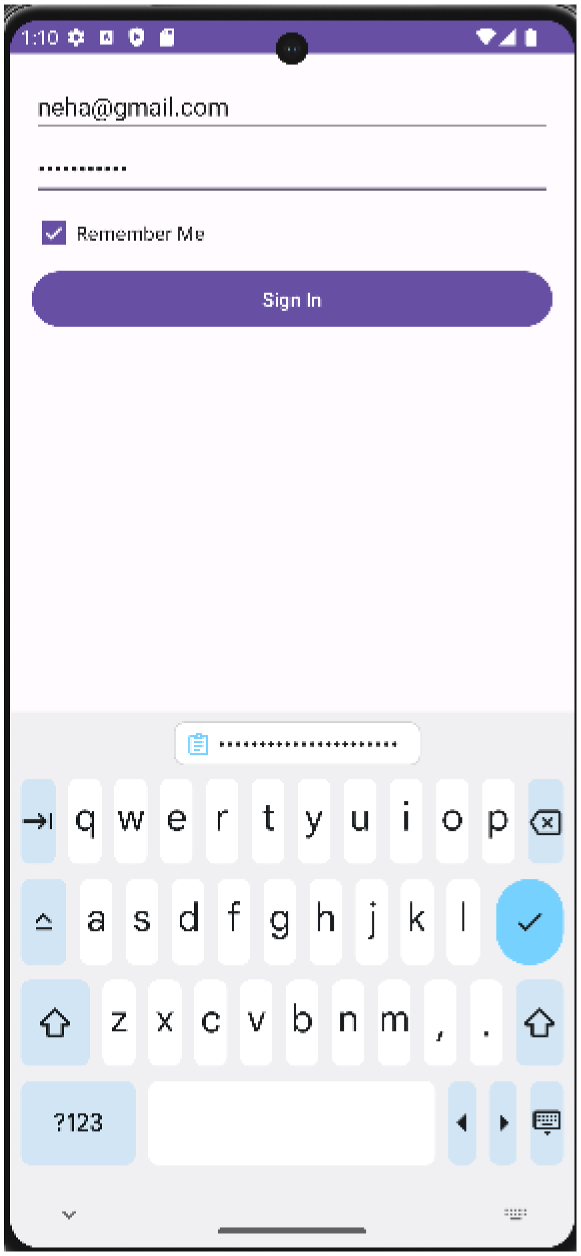
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/welcomeText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="24sp"
        android:text="Welcome!" />

    <Button
        android:id="@+id/logoutButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Logout" />

</LinearLayout>
```

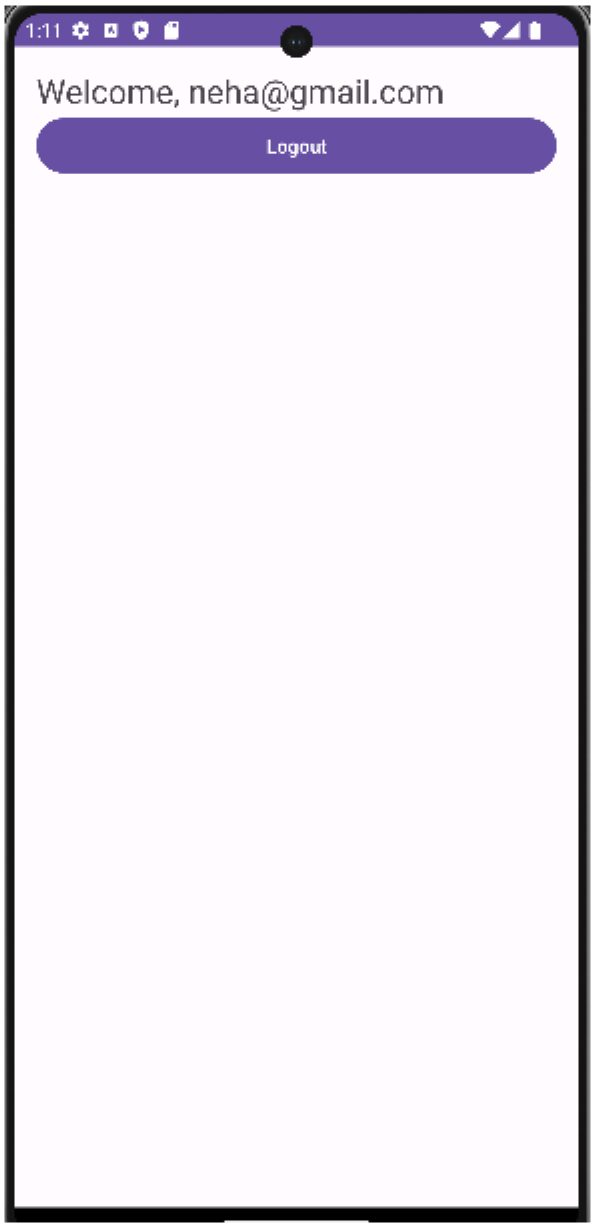
AndroidManifest.xml (nothing to add)

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

  <application
      android:allowBackup="true"
      android:dataExtractionRules="@xml/data_extraction_rules"
      android:fullBackupContent="@xml/backup_rules"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:supportsRtl="true"
      android:theme="@style/Theme.Practical05"
      tools:targetApi="31">
      <activity
          android:name=".HomeActivity"
          android:exported="false" />
      <activity
          android:name=".MainActivity"
          android:exported="true">
          <intent-filter>
              <action android:name="android.intent.action.MAIN" />

              <category android:name="android.intent.category.LAUNCHER" />
          </intent-filter>
      </activity>
  </application>

</manifest>
```

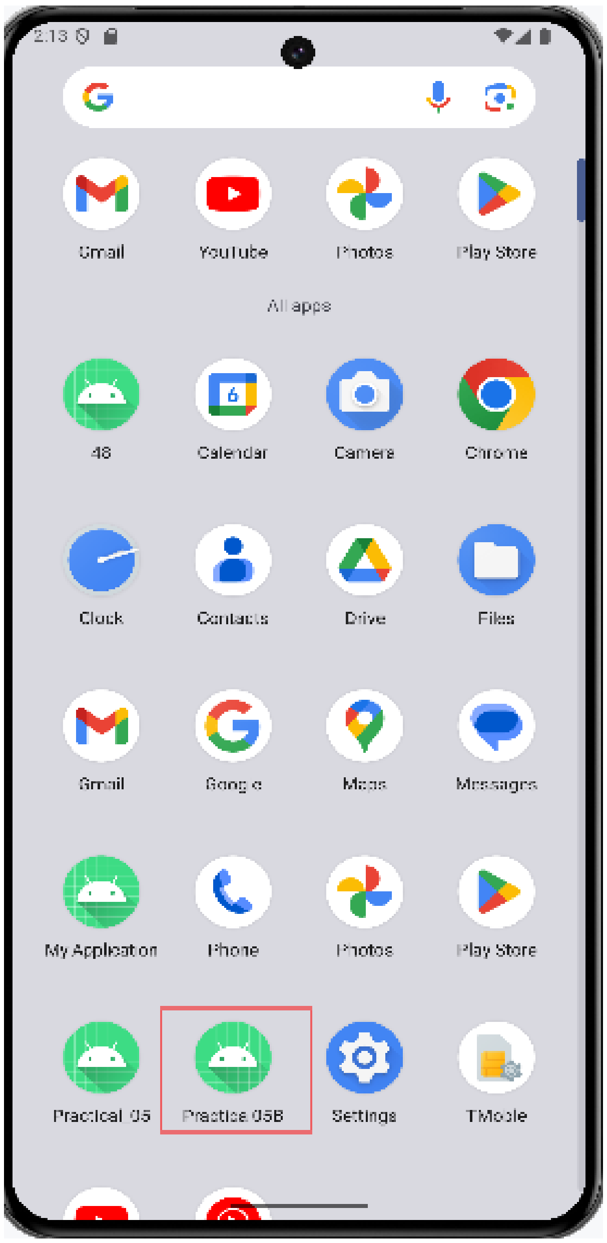
OUTPUT:

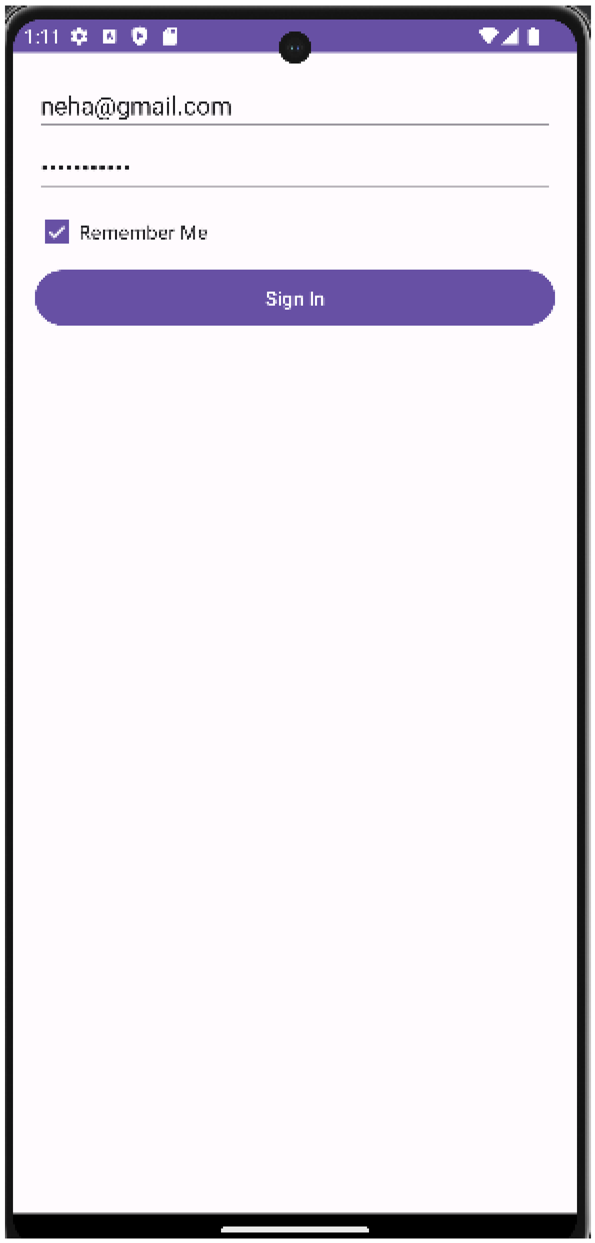

1:09
Email
Password
Remember Me
Sign In


1:10
neha@gmail.com
....
Remember Me
Sign In
?123



Go back and open the app again. To go back just swipe to the right.
There will be an application with the name of your file





CONCLUSION:
Implementing file I/O in Android allows you to manage data persistently through operations such as creating, writing, reading, and deleting files within the app's internal storage. Utilizing Shared Preferences enables efficient storage and retrieval of simple key-value pairs, such as user login credentials and settings. Together, these techniques facilitate robust data management and enhance user experience by maintaining state and preferences across application sessions.