Experiment 3

```
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
```

```
public class MainActivity extends AppCompatActivity {

   private static final int PERMISSION_REQUEST_CODE = 1;
   EditText phoneNumberEditText, messageEditText;

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

       phoneNumberEditText = findViewById(R.id.editTextPhoneNumber);
       messageEditText = findViewById(R.id.editTextMessage);
       Button sendButton = findViewById(R.id.buttonSend);

       sendButton.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               String phone  = phoneNumberEditText.getText().toString();
               String message = messageEditText.getText().toString();
               if(phone.isEmpty() || message.isEmpty()) {
                   Toast.makeText(MainActivity.this, "Please fill all the fields",
Toast.LENGTH_SHORT).show();
               }
               else {
                   if(checkPermission())
                       sendSMS(phone, message);
                   else
                       requestPermission();
               }
           }
       });
   }
   public boolean checkPermission() {
       int result = ActivityCompat.checkSelfPermission(this,
Manifest.permission.SEND_SMS);
       return result == PackageManager.PERMISSION_GRANTED;
   }

   public void requestPermission() {
       ActivityCompat.requestPermissions(this, new String[]
```

```
{Manifest.permission.SEND_SMS}, PERMISSION_REQUEST_CODE);
    }

    public void sendSMS(String phone, String message) {
        try {
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(phone, null, message, null, null);
            Toast.makeText(MainActivity.this, "Message sent successfully",
Toast.LENGTH_SHORT).show();
        }
        catch (Exception e) {
            Toast.makeText(MainActivity.this, "An error has occurred",
Toast.LENGTH_SHORT).show();
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if(requestCode == PERMISSION_REQUEST_CODE) {
            if(grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED)
                Toast.makeText(MainActivity.this, "Permission Granted",
Toast.LENGTH_SHORT).show();
            else
                Toast.makeText(MainActivity.this, "Permission Denied",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

Experiment 4A

```
public class MainActivity extends AppCompatActivity {
    EditText emailField, passwordField;
    Button loginButton;

    private static final String USERNAME = "admin@gmail.com", PASSWORD = "admin";

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

        emailField = findViewById(id.email);
```

```
        passwordField = findViewById(R.id.password);
        loginButton = findViewById(R.id.login);

        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String email = emailField.getText().toString();
                String password = passwordField.getText().toString();

                if(email.isEmpty() || password.isEmpty())
                    Toast.makeText(MainActivity.this, "Please fill all fields",
Toast.LENGTH_SHORT).show();
                else
                    authenticateUser(email, password);
            }
        });

    }
    protected void authenticateUser(String email, String password) {
        if(email.equals(USERNAME) && password.equals(PASSWORD)) {
            Toast.makeText(MainActivity.this, "Successfully Authenticated",
Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(MainActivity.this, HomeActivity.class);
            startActivity(intent);
            finish();
        }
        else
            Toast.makeText(MainActivity.this, "Invalid Credentials",
Toast.LENGTH_SHORT).show();

    }
}
```

Experiment 4B

```
public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "user_database.db";
    private static final int DATABASE_VERSION = 1;

    private static final String TABLE_USERS = "users";
    private static final String COLUMN_EMAIL = "username";
    private static final String COLUMN_PASSWORD = "password";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    public boolean checkUser(String email, String password) {
```

```
        SQLiteDatabase db = this.getReadableDatabase();
        String query = "SELECT * FROM " + TABLE_USERS + " WHERE "
                + COLUMN_EMAIL + " = ? AND "
                + COLUMN_PASSWORD + " = ?";
        Cursor cursor = db.rawQuery(query, new String[]{email, password});
        boolean exists = cursor.getCount() > 0;
        cursor.close();
        db.close();
        return exists;
    }
}
```

Experiment 5

```
implementation 'com.google.android.material.material.1.3.0-alpha3'
```

navigation_menu.xml

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

      <item
            android:id="@+id/nav_account"
            android:title="My Account" />

      <item
            android:id="@+id/nav_settings"
            android:title="Settings" />

      <item
            android:id="@+id/nav_logout"
            android:title="Logout" />

</menu>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>

<!-- the root view must be the DrawerLayout -->

<androidx.drawerlayout.widget.DrawerLayout
      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:id="@+id/my_drawer_layout"
	android:layout_width="match_parent"
	android:layout_height="match_parent"
	tools:context=".MainActivity"
	tools:ignore="HardcodedText">

	<LinearLayout
		android:layout_width="match_parent"
		android:layout_height="match_parent">

		<TextView
			android:layout_width="match_parent"
			android:layout_height="wrap_content"
			android:layout_marginTop="128dp"
			android:gravity="center"
			android:text="GeeksforGeeks"
			android:textSize="18sp" />
	</LinearLayout>

	<!-- this the navigation view which draws and shows the navigation drawer
-->
	<!-- include the menu created in the menu folder -->
	<com.google.android.material.navigation.NavigationView
		android:layout_width="wrap_content"
		android:layout_height="match_parent"
		android:layout_gravity="start"
		app:menu="@menu/navigation_menu" />

</androidx.drawerlayout.widget.DrawerLayout>
```

app/res/values/string.xml

```
<resources>
	<string name="app_name">Navigation Drawer</string>
	<!-- to toggle the open close button of the navigation drawer -->
	<string name="nav_open">Open</string>
	<string name="nav_close">Close</string>
</resources>
```

MainActivity.java

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.drawerlayout.widget.DrawerLayout;
import android.os.Bundle;
```

```
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

	public DrawerLayout drawerLayout;
	public ActionBarDrawerToggle actionBarDrawerToggle;

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

		// drawer layout instance to toggle the menu icon to open
		// drawer and back button to close drawer
		drawerLayout = findViewById(R.id.my_drawer_layout);
		actionBarDrawerToggle = new ActionBarDrawerToggle(this, drawerLayout,
R.string.nav_open, R.string.nav_close);

		// pass the Open and Close toggle for the drawer layout listener
		// to toggle the button
		drawerLayout.addDrawerListener(actionBarDrawerToggle);
		actionBarDrawerToggle.syncState();

		// to make the Navigation drawer icon always appear on the action bar
		getSupportActionBar().setDisplayHomeAsUpEnabled(true);
	}

	// override the onOptionsItemSelected()
	// function to implement
	// the item click listener callback
	// to open and close the navigation
	// drawer when the icon is clicked
	@Override
	public boolean onOptionsItemSelected(@NonNull MenuItem item) {

		if (actionBarDrawerToggle.onOptionsItemSelected(item)) {
			return true;
		}
		return super.onOptionsItemSelected(item);
	}
}
```

Experiment 10

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

```
public class MainActivity extends AppCompatActivity {
   WebView mywebView;

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

       mywebView = findViewById(R.id.webview);

       mywebView.setWebViewClient(new WebViewClient());
       mywebView.loadUrl("https://innovatia23.tech");
       WebSettings webSettings= mywebView.getSettings();
       webSettings.setJavaScriptEnabled(true);
   }
}
```

Experiment 11

```
public class MainActivity extends AppCompatActivity {
   private EditText editTextAddress; private
   EditText editTextSubject; private
   EditText editTextMessage;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);

       editTextAddress = findViewById(R.id.edit_text_address);
       editTextSubject = findViewById(R.id.edit_text_subject);
       editTextMessage = findViewById(R.id.edit_text_message);
       Button sendEmailButton = findViewById(R.id.send_email_button);

       sendEmailButton.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View v) {
               sendEmail();
           }
       });
   }
   private void sendEmail() {
       String[] recipientList = {editTextAddress.getText().toString()};
       String subject = editTextSubject.getText().toString();
       String message = editTextMessage.getText().toString();
       Intent intent = new Intent(Intent.ACTION_SENDTO);
       intent.setData(Uri.parse("mailto:"));
```

```
        intent.putExtra(Intent.EXTRA_EMAIL, recipientList);
        intent.putExtra(Intent.EXTRA_SUBJECT, subject);
        intent.putExtra(Intent.EXTRA_TEXT, message);
        if (intent.resolveActivity(getPackageManager()) != null) {
            startActivity(Intent.createChooser(intent, "Choose an Email client"));
        }
    }
}
```

Experiment 8

```
protected void authenticateUser(String email, String password) {
    if(email.equals(USERNAME) && password.equals(PASSWORD)) {
        Toast.makeText(MainActivity.this, "Successfully Authenticated",
Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(MainActivity.this, HomeActivity.class);
        intent.putExtra("name", email);
        startActivity(intent);
        finish();
    }
    else
        Toast.makeText(MainActivity.this, "Invalid Credentials",
Toast.LENGTH_SHORT).show();

}
```

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

    welcomeText = findViewById(R.id.welcome);

    Intent intent = getIntent();
    String name = intent.getStringExtra("name");

    welcomeText.setText("Welcome to our application, " + name);
}
```

Experiment 7

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);

        NotificationHelper.sendNotification(MainActivity.this, "Hello World", "This
is an example notification message");
    }
}
```

```
public class NotificationHelper {

    private static final String CHANNEL_ID = "MyChannelID";
    private static final int NOTIFICATION_ID = 123;

    public static void sendNotification(Context context, String title, String
message) {
        NotificationManager notificationManager =
                (NotificationManager)
context.getSystemService(Context.NOTIFICATION_SERVICE);

        // Create an explicit intent for the MainActivity
        Intent intent = new Intent(context, MainActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
        PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, intent,
PendingIntent.FLAG_IMMUTABLE);

        // Build the notification
        NotificationCompat.Builder builder = new NotificationCompat.Builder(context,
CHANNEL_ID)
                .setSmallIcon(R.drawable.ic_notification)
                .setContentTitle(title)
                .setContentText(message)
                .setPriority(NotificationCompat.PRIORITY_DEFAULT)
                .setContentIntent(pendingIntent)
                .setAutoCancel(true);

        // Show the notification
        notificationManager.notify(NOTIFICATION_ID, builder.build());
    }
}
```