Name of Student : Mohammed Mohtashim Ansari			
Roll Number : 1		**Practical Number: 3**	
Title of LAB Assignment : Android program based on intents.(Implicit intents).			
DOP : 02-09-2024		**DOS : 2-09-2024**	
CO Mapped : **CO1**	**PO Mapped:** **PO2, PO5, PSO1**	**Faculty Signature:**	**Marks :**

Theory of Intents in Android

Intents in Android are a fundamental concept used to facilitate communication between components, such as activities, services, and broadcast receivers. They allow you to perform actions like starting activities, sending data between activities, starting services, and broadcasting messages to other applications.

1. Types of Intents

- **Explicit Intents**: These are used when you know the specific component (activity, service, etc.) you want to start. Explicit intents specify the target component directly using the class name.
    - Example:
      Java
    -

2. Intent Structure

- **Action**: A string that specifies the desired action (e.g., `Intent.ACTION_VIEW`, `Intent.ACTION_SEND`, etc.).
- **Data**: The data associated with the intent, often in the form of a URI.
- **Category**: An optional string that provides additional information about the action being performed (e.g., `Intent.CATEGORY_DEFAULT`).
- **Extras**: A `Bundle` of key-value pairs providing additional information to the component handling the intent.
- **Component**: The name of the component that should handle the intent, used in explicit intents.
- **Flags**: Additional instructions on how the intent should be handled (e.g., `Intent.FLAG_ACTIVITY_NEW_TASK`).

3. Common Intent Actions

- **Intent.ACTION_VIEW**: Used to display data to the user. For example, opening a URL in a browser or showing a location on a map.
- **Intent.ACTION_SEND**: Used to send data from one activity to another. Typically used for sharing content (e.g., sharing a text message, image, etc.).
- **Intent.ACTION_DIAL**: Used to open the dialer with a number pre-filled.
- **Intent.ACTION_CALL**: Used to make a call directly (requires permission).
- **Intent.ACTION_EDIT**: Used to edit a specific piece of data.

4. Resolving Intents

- The Android system matches an implicit intent with one or more components by comparing the intent's action, data, and category with the `intent-filters` declared in the manifest file of other applications.
- **Intent Resolution**: When an implicit intent is broadcast, the system searches for components that can handle the intent. If multiple components can handle the intent, the system presents a chooser dialog to the user.

5. Using Intents for Data Transfer

- **Extras**: Data can be passed between components using the `putExtra()` method. For example:
- The receiving activity can retrieve the data using `getIntent().getStringExtra("key");`.

6. Pending Intents

- A `PendingIntent` is a token that you give to another application (e.g., a notification manager or alarm manager) which allows the application to execute a predefined action as if it were your application, even if your application is no longer running.

Summary

Intents are a powerful mechanism in Android for inter-component communication, facilitating the interaction between different parts of an app and even different apps. They allow for both direct (explicit) and flexible (implicit) invocation of activities, services, and broadcasts. Understanding how to create, manipulate, and resolve intents is key to developing robust and interactive Android applications.

Android uses Intent for communicating between the components of an Application and also from one application to another application.

Intents are the objects which are used in android for passing the information among Activities in an Application and from one app to another also.

Types of Intents:
Intent are of two types: Explicit Intent and Implicit Intent

Types of Intents
Explicit Intent:

Explicit Intents are used to connect the application internally.
In Explicit we use the name of the component which will be affected by Intent. For Example: If we know the class name then we can navigate the app from One Activity to another activity using Intent. In the similar way we can start a service to download a file in the background process.
Explicit Intent works internally within an application to perform navigation and data transfer.

```
Intent intent = new Intent(getApplicationContext(), SecondActivity.class);
startActivity(intent);
```

Implicit Intent:
In Implicit Intents we do need to specify the name of the component. We just specify the Action which has to be performed and further this action is handled by the component of another application.
The basic example of implicit Intent is to open any web page

```
Intent intentObj = new Intent(Intent.ACTION_VIEW);
intentObj.setData(Uri.parse("https://www.abhiandroid.com"));
startActivity(intentObj);
```

Intent requests
In order to launch Google Maps with an intent you must first create an Intent object, specifying its action, URI and package.

Action: All Google Maps intents are called as a View action — ACTION_VIEW.
URI: Google Maps intents use URL encoded that specify a desired action, along with some data with which to perform the action.
Package: Calling setPackage("com.google.android.apps.maps") will ensure that the Google Maps app for Android handles the Intent. If the package isn't set, the system will determine which apps can handle the Intent. If multiple apps are available, the user may be asked which app they would like to use.

URL encoded query strings
All strings passed to the Google Maps Intents must be URI encoded. For example, the string "1st & Pike, Seattle" should become 1st%20%26%20Pike%2C%20Seattle. Spaces in the string can be encoded with %20 or replaced with the plus sign (+).

You can use the android.net.Uri parse() method to encode your strings. For example:

Uri gmmIntentUri = Uri.parse("geo:37.7749,-122.4192?q=" + Uri.encode("1st & Pike, Seattle"));

Uri gmmIntentUri = Uri.parse("geo:0,0?q=1600 Amphitheatre Parkway, Mountain+View, California");
Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
mapIntent.setPackage("com.google.android.apps.maps");
startActivity(mapIntent);

open up a dialog which allow you to pick a map app to show the address you passed in the intent.

```
Intent intent = new Intent(android.content.Intent.ACTION_VIEW,
      Uri.parse("geo:0,0?q=replace+this+with+an+address"));

startActivity(intent);
```

This Android code snippet will launch the google map with direction to the address you passed in.

```
Intent intent = new Intent(Intent.ACTION_VIEW,
   Uri.parse("google.navigation:q=replace+this+with+an+address"));

startActivity(intent);
```

Passing in the latitude,longitude to the map app.

```
Intent intent = new Intent(android.content.Intent.ACTION_VIEW,
      Uri.parse("geo:46.7170627,-71.2884537"));
startActivity(intent);
```

Opening a Website

```
String url = editText.getText().toString();
Intent urlIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
startActivity(urlIntent);
```

Displaying a particular location

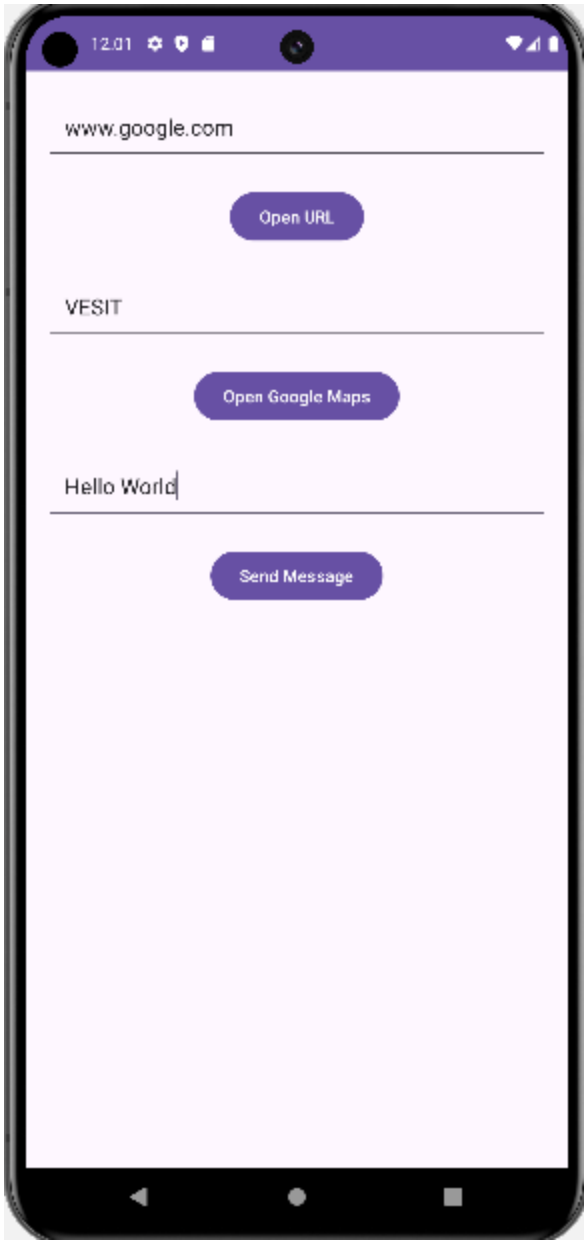
```
String map = editmap.getText().toString();
Uri gmmIntentUri = Uri.parse("geo:0,0?q="+map);
Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
mapIntent.setPackage("com.google.android.apps.maps");
startActivity(mapIntent);
```

Sharing Text message

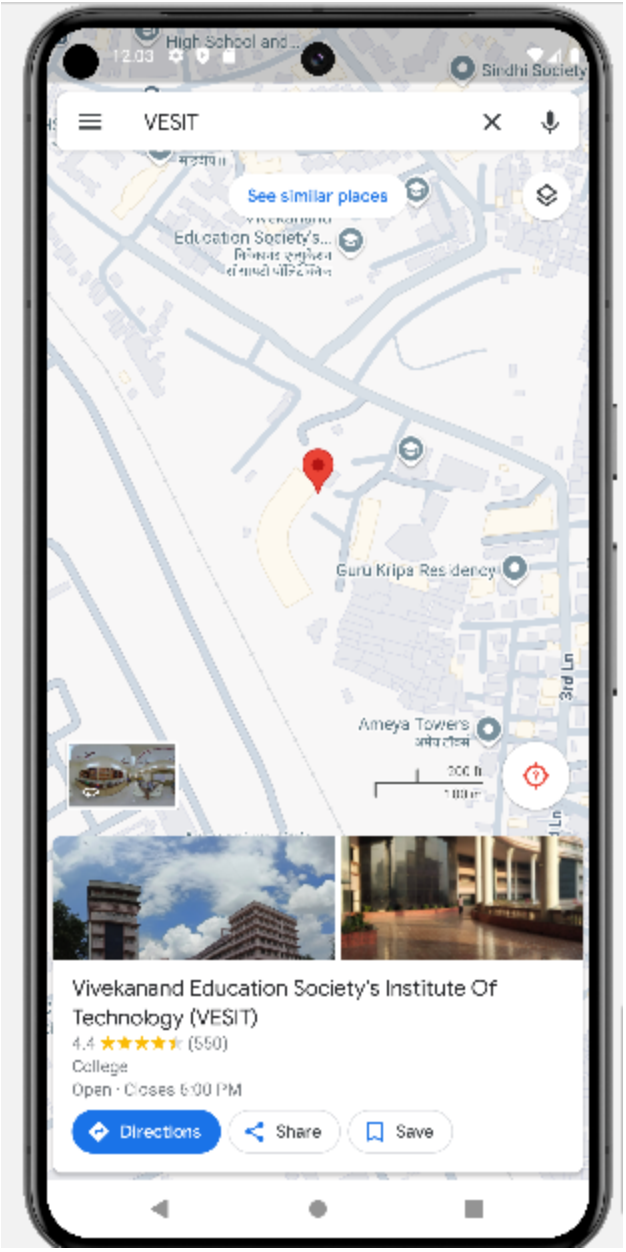
```
String textMsg = textMessage.getText().toString();
Intent intent2 = new Intent();
intent2.setAction(Intent.ACTION_SEND);
intent2.setType("text/plain");
intent2.putExtra(Intent.EXTRA_TEXT, textMsg );
startActivity(Intent.createChooser(intent2, "Share via"));
```

OUTPUT
Open URL

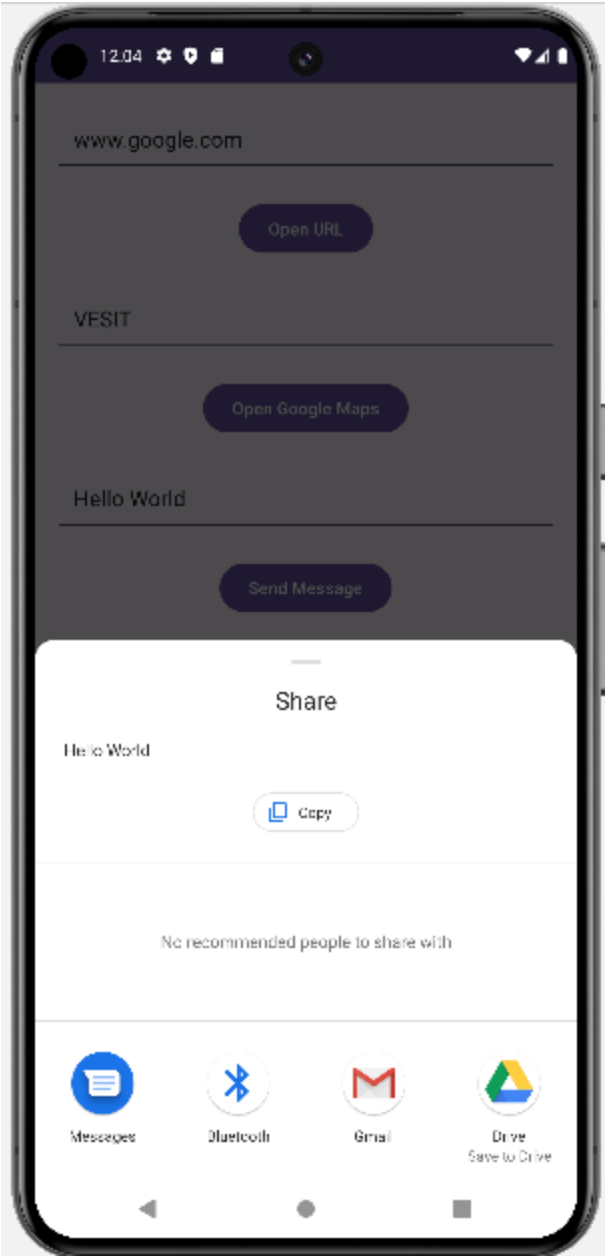

12:07
google.com
ALL
IMAGES
Sign In
GOOGLE
Google offered in:


12:01
www.google.com
Open URL
VESIT
Open Google Maps
Hello World
Send Message

Open Maps


VESIT
See similar places
Guru Kripa Residency
Ameya Towers
Vivekanand Education Society's Institute Of Technology (VESIT)
4.4 (550)
College
Open · Closes 6:00 PM
Directions
Share
Save

Send Msg


12:04
www.google.com
Open URL
VESIT
Open Google Maps
Hello World
Send Message
Share
Hello World
Copy
No recommended people to share with
Messages
Bluetooth
Gmail
Drive
Save to Drive