

# ***MOBILE COMPUTING LAB***

***ETIT-452***

Faculty name: Mrs. Divya Arora

Student name: Syeda Reeha Quasar

Roll No.: 14114802719

Semester: 8<sup>th</sup>

Group: 8C7



Maharaja Agrasen Institute of Technology, PSP Area,

Sector – 22, Rohini, New Delhi – 110085

## Index

Exp No	Experiment Name	Date of performance	Date of checking	Marks	Sign.
1.	Write a WML program to print formatted Text on the mobileScreen using various tags.				
2.	Write a WML program to connect multiple cards from same desk.				
3.	Write WML program to display table with three columns Image name, Image and third column contain hyperlink to open another card.				
4.	Write a WML program to create a form with multiple options.				
5.	Write a WML program to use the time control and to trigger on pick event				
6.	Write a WML script to find maximum out of two numbers with help of inbuilt function Lang.Max() and to find absolute value with help of inbuilt function Lang.abs()				
7.	Write a Program in NS3 to Simulate OLSR				
8.	Write a Program in NS3 to Simulate AODV				
9.	Write an Android Program to create list view, grid view and database connectivity.				
10.	Make an application using XML or Android from the following: Game, Clock, calendar, Converter, phone book, Text Editor.				
<b>Beyond the Syllabus Questions</b>					
1.	Developing Android App such as E-Commerce based or E learning Based apps.				

<b>2.</b>	Program to develop a calling application.(Android)				
<b>3.</b>	Program to develop a mailing application. (Android)				
<b>4.</b>	Program to design a contact application. (Android)				

## Experiment 1

**Aim :** Write a WML program to print a formatted Text on the mobile Screen using various tags.

**Theory :**

Attribute	Value	Description
align	left right center	Aligns the paragraph. Default is "left"
mode	wrap nowrap	Sets whether a paragraph should wrap lines or not.
xml:lang	language_code	Sets the language used in the element
class	cdata	Sets a class name for the element. The class name is case sensitive. An element can be connected to multiple classes. Multiple class names within the class Attribute are separated by white space
id	id	Sets a unique name for the element

WML Elements	
<b>	Defines bold text
<big>	Defines big text
<em>	Defines emphasized text
<i>	Defines italic text
<small>	Defines small text
<strong>	Defines strong text
<u>	Defines underlined text
	<b>Purpose</b>

<!-->	Defines a WML comment
<wml>	Defines a WML deck (WML root)
<head>	Defines head information
<meta>	Defines meta information
<card>	Defines a card in a deck
<access>	Defines information about the access control of a deck
<template>	Defines a code template for all the cards in a deck

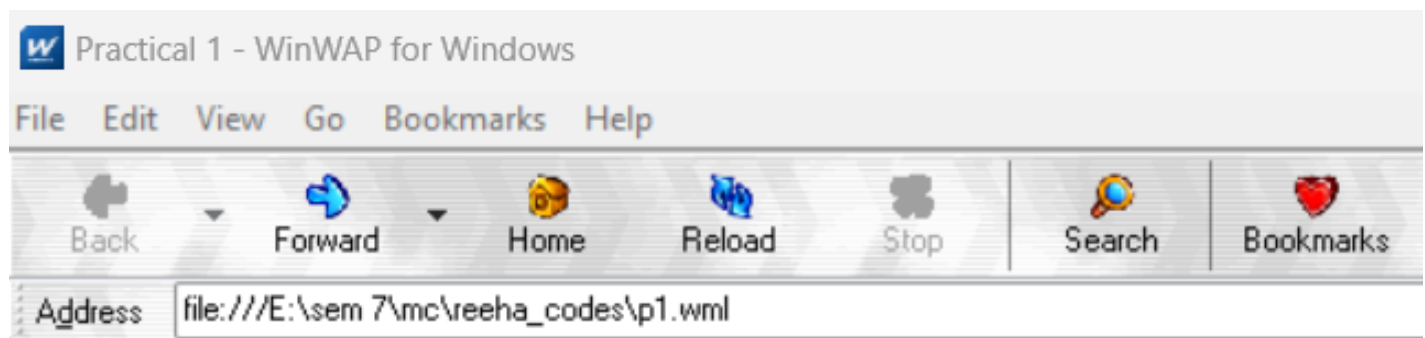
### Code :

```

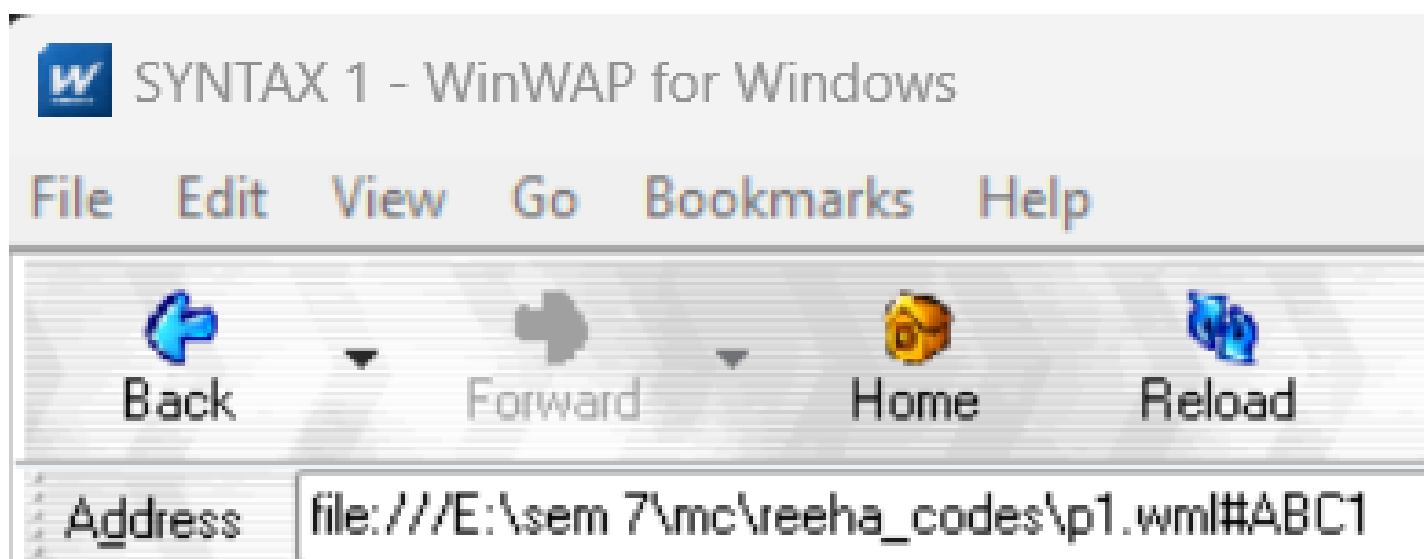
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
<wml>
<card id="ABC" title="Practical 1">
<p>
Syeda Reeha Quasar 14114802719
</p>
<onevent type="ontimer">
<go href="#ABC1"/>
</onevent>
<timer value="50"/>
</card>
<card id="ABC1" title="SYNTAX 1">
<p>
<b>Syeda Reeha Quasar</b><br>
<big>Syeda Reeha Quasar</big><br>
<em>Syeda Reeha Quasar</em><br>
<i>Syeda Reeha Quasar</i><br>
<small>Syeda Reeha Quasar</small><br>
<strong>Syeda Reeha Quasar</strong><br>
<u>Syeda Reeha Quasar</u>
</p>
</card>
</wml>

```

Output :



Syeda Reeha Quasar 14114802719



**Syeda Reeha Quasar**  
Syeda Reeha Quasar  
*Syeda Reeha Quasar*  
*Syeda Reeha Quasar*  
Syeda Reeha Quasar  
**Syeda Reeha Quasar**  
Syeda Reeha Quasar

## VIVA QUESTIONS

### Q1. What is the use of WML decks?

WML decks provide following functionalities:

- WML document used for creating an application called a deck that provides way to add more pages.
- Decks can be used to insert the data into one or more cards or pages.
- Deck interacts with the user and the framework on which the application is being built.
- Decks are stored on a configured web server whose function is to include data of MIME type.

### Q2. What is WSDL?

WSDL stands for Web Services Description Language. WSDL is an XML-based protocol for information exchange in decentralized and distributed environments. It describes how to access a web service and what operations it will perform. WSDL is a language for describing how to interface with XML-based services.

### Q3. What are WML variables? How to use them?

Variables are named storage areas that can be manipulated.

Variable expansion occurs at runtime, in the micro-browser or emulator. This means it can be concatenated with or embedded in other text. Variables are referenced with a preceding dollar sign, and any single dollar sign in your WML deck is interpreted as a variable reference.

You can set the value of a variable in the following ways:

- Using the <setvar/> tag
- Using data collection tags <select> and <input/>
- Using the setVar() function of WMLScript's WMLBrowser standard library

### Q4. How can we refresh card variables?

The refresh() function, as suggested by its function name, is used to refresh the current card on the WML browser.

## Experiment 2

**Aim :** Write a WML program to connect multiple cards from same desk.

**Theory :**

### Introduction about WML Tags:

The <anchor>...</anchor> tag pair is used to create an anchor link. It is used together with other WML elements called <go/>, <refresh> or <prev/>. These elements are called task elements and tell WAP browsers what to do when a user selects the anchor link

You can enclose Text or image along with a task tag inside <anchor>...</anchor> tag pair.

### Attributes:

This element supports the following attributes:

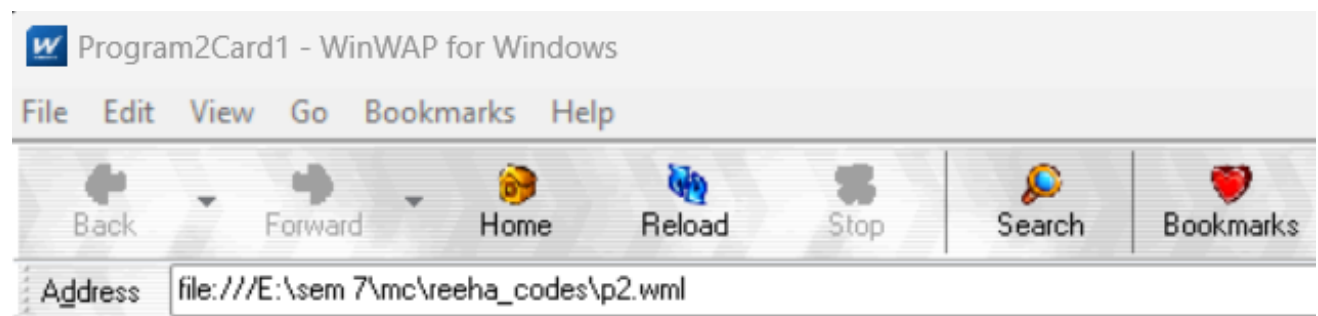
Attribute	Value	Description
title	cdata	Defines a text identifying the link
xml:lang	language_code	Sets the language used in the element
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

**Code :**

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
<wml>
<card id="CardNo.1" title="Program2Card1">
<p>You Are Visiting Card 1</p>
<do type="accept" name="CardNo.2" label="Continue">
<go href="#CardNo.2"/>
</do>
</card>
<card id="CardNo.2" title="Program2Card2">
<p>You Are Visiting Card 2</p>
<do type="accept" name="CardNo.3" label="Continue">
<go href="#CardNo.3"/>
</do>
</card>
<card id="CardNo.3" title="Program2Card3">
<p>You Are Visiting Card 3</p>
<do type="accept" name="CardNo.1" label="Go Back">
<go href="#CardNo.1"/>
</do>
</card>
</wml>
```

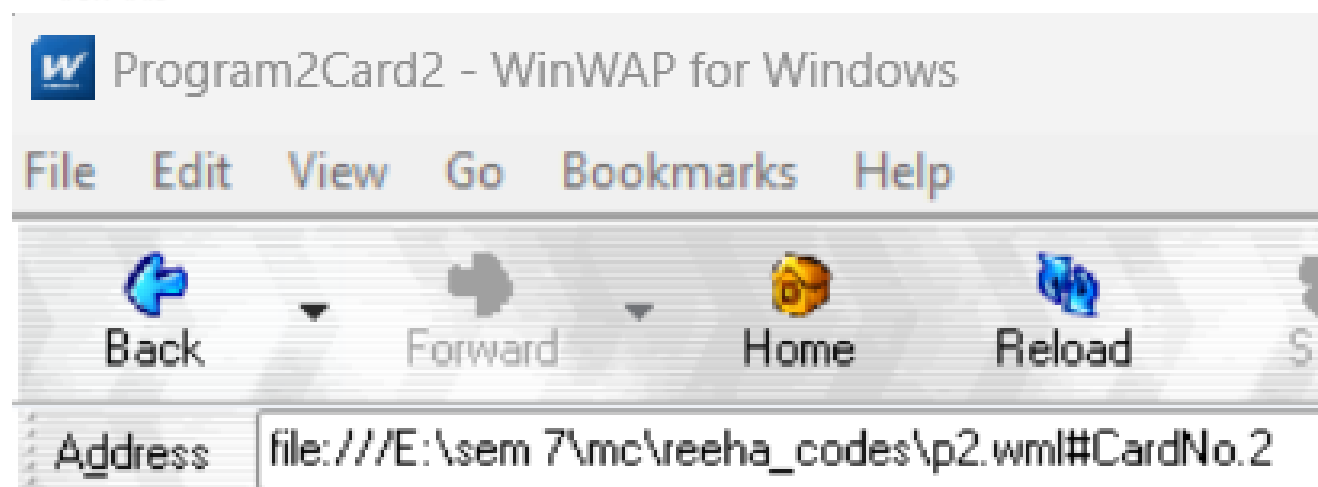


Output :



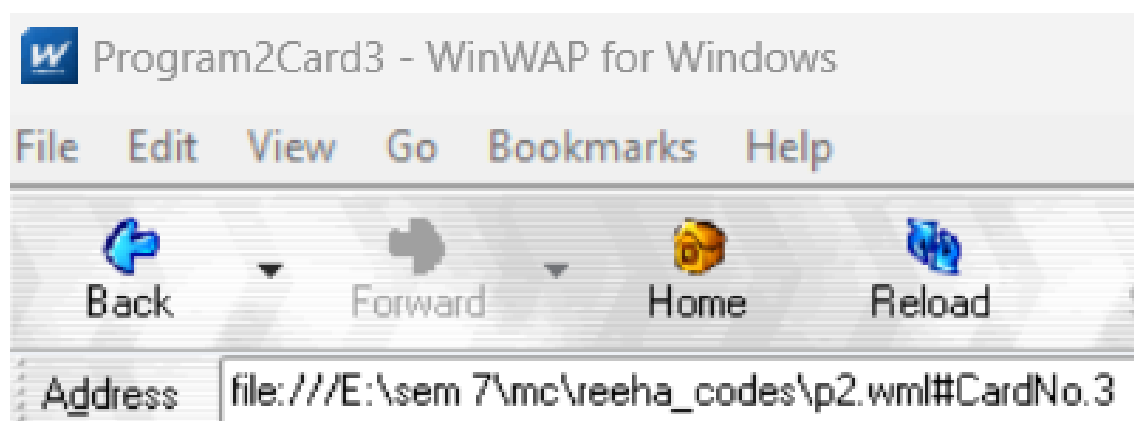
You Are Visiting Card 1

[Continue](#)



You Are Visiting Card 2

[Continue](#)



You Are Visiting Card 3

[Go Back](#)

## **VIVA QUESTIONS**

### **Q1. What is the use of XML?**

- XML can keep data separated from your HTML
- XML can be used to store data inside HTML documents
- XML can be used as a format to exchange information
- XML can be used to store data in files or in databases

### **Q2. What is Meta data?**

Metadata is data that describes other data. Meta is a prefix that -- in most information technology usages -- means "an underlying definition or description." Metadata summarizes basic information about data, which can make finding and working with particular instances of data easier.

### **Q3. What is the difference between HTML and WML?**

A main difference between HTML and WML is that the basic unit of navigation in HTML is a page, while that in WML is a card.

WML is used on phones while HTML is used for desktop client HTML requires a lot more processing power than WML

### **Q4. What is XML DOM Document?**

The XML DOM is a standard object model for XML, a standard programming interface for XML, platform- and language independent. The XML DOM is a standard for how to get, change, add, or delete XML elements.

## Experiment 3

**Aim :** Write WML program to display table with three columns Image name, Image and third column contain hyperlink to open another card.

### Theory :

The element `<table>` along with `<tr>` and `<td>` is used to create a table in WML. WML does not allow the nesting of tables.

A `<table>` element should be put with-in `<p> ... </p>` elements.

Attributes:

The element supports the following attributes:

Attribute	Value	Description
columns	number	Sets the number of columns in the table
align	L C R	To specify the horizontal text alignment of the columns, you need to assign three letters to the align attribute. Each letter represents the horizontal text alignment of a column. The letter can be L, C, or R. For example, if you want the following settings to be applied to your table: First table column -- Left-aligned Second table column -- Center-aligned Third table column -- Right-aligned Then you should set the value of the align attribute to LCR.
xml:lang	language_code	Sets the language used in the element
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

### Code :

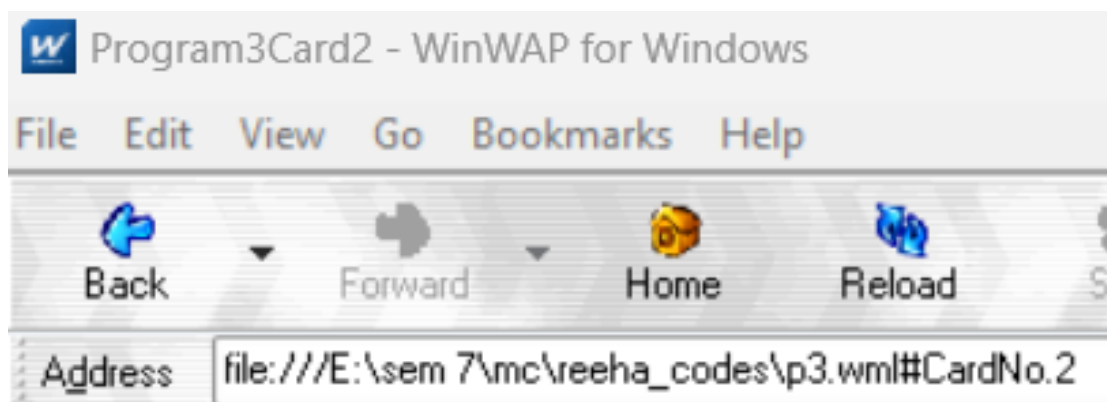
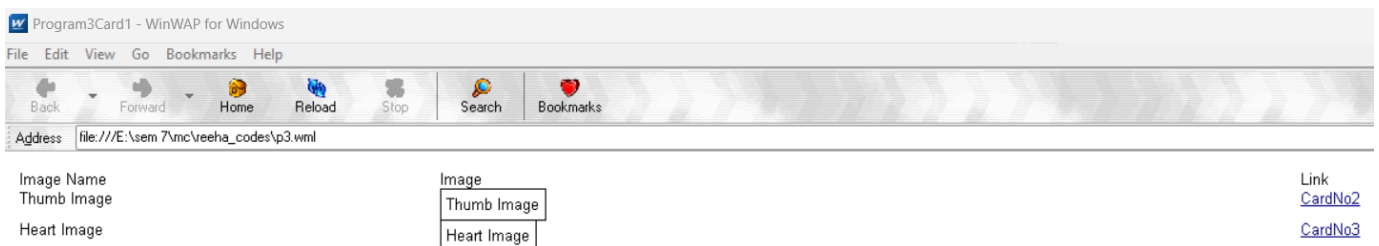
```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
<wml>
<card id="CardNo.1" title="Program3Card1">
<p>
<table columns="3" align="LCR">
<tr>
<td>Image Name</td>
<td>Image</td>
<td>Link</td>
</tr>
<tr>
<td>Thumb Image</td>
<td></td>
<td><anchor><a href="#CardNo.2">CardNo2</a></anchor></td>
</tr>
</tr>
```

```

<td>Heart Image</td>
<td></td>
<td><anchor><a href="#CardNo.3">CardNo3</a></anchor></td>
</tr>
</table>
</p>
</card>
<card id="CardNo.2" title="Program3Card2">
<p>You Are Visiting Card 2</p>
<do type="accept" name="CardNo.3" label="Continue">
<go href="#CardNo.3"/>
</do>
</card>
<card id="CardNo.3" title="Program3Card3">
<p>You Are Visiting Card 3</p>
<do type="accept" name="CardNo.1" label="Go Back">
<go href="#CardNo.1"/>
</do>
</card>
</wml>

```

## Output :



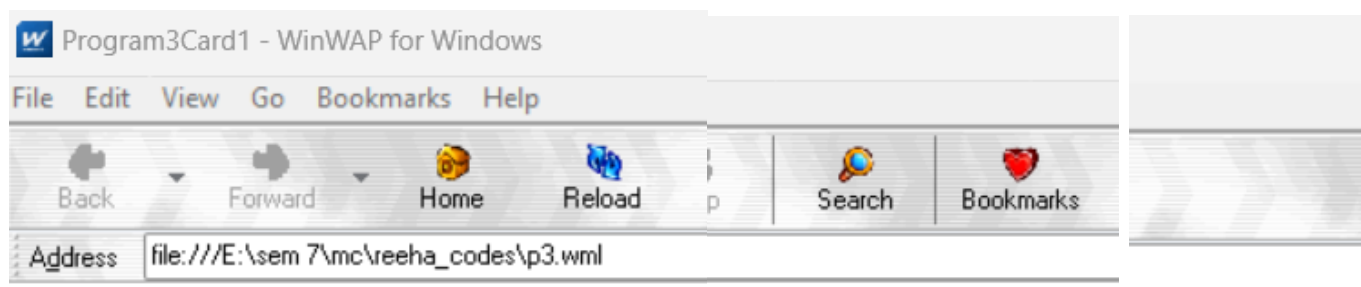
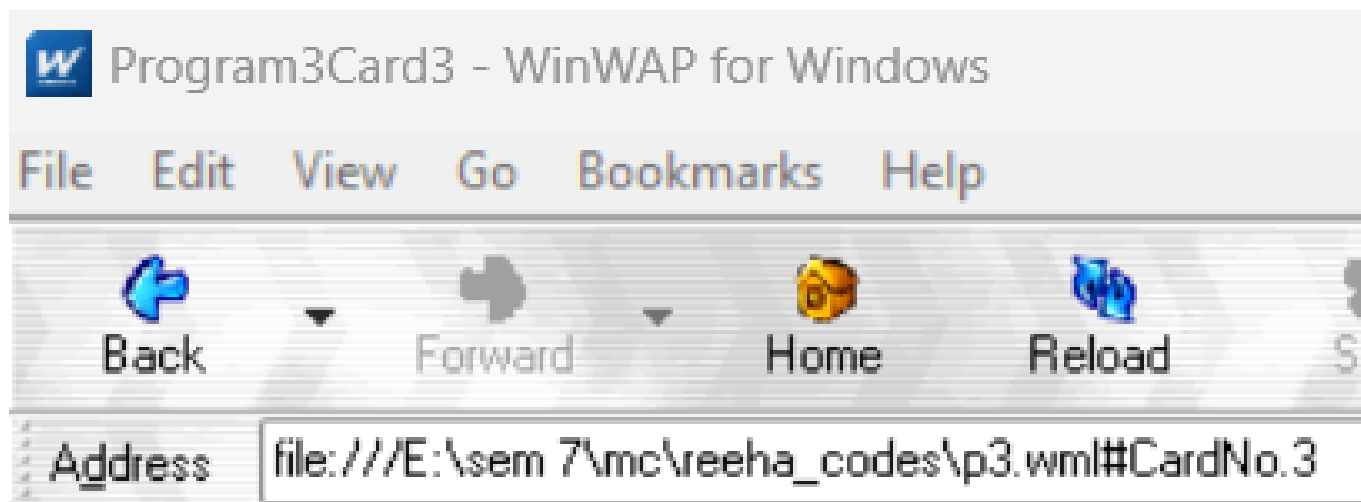


Image Name
Thumb Image
Heart Image

Image
Thumb Image
Heart Image

Link
<a href="#">CardNo2</a>
<a href="#">CardNo3</a>

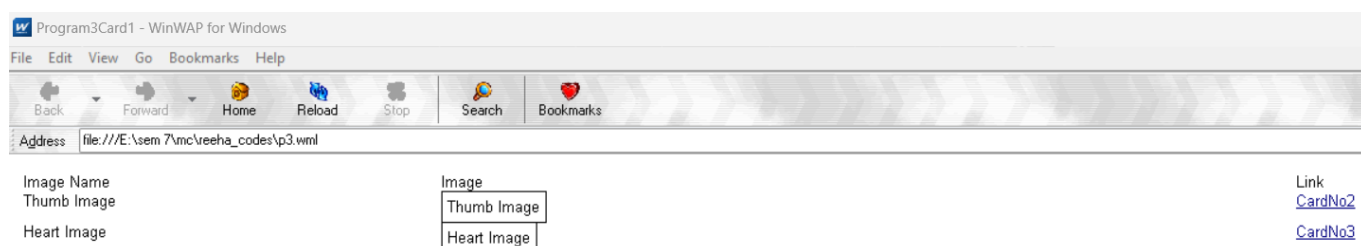


Image Name
Thumb Image
Heart Image

Image
Thumb Image
Heart Image

Link
<a href="#">CardNo2</a>
<a href="#">CardNo3</a>

## Code:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="text" title="059 Exp-3">
<p><b>Image Table</b></p>
<p>
<table columns="3" align="LCR">
<tr>
<td><b>Image name</b></td>
<td><b>Image</b></td>
<td><b>Image link</b></td>
</tr>
<tr>
<td>Sun</td>
<td></td>
<td><anchor>Image Link<go href="sun.wml"/></anchor></td>
</tr>
<tr>
<td>Earth</td>
<td></td>
<td><anchor>Image Link<go href="earth.wml"/></anchor></td>
</tr>
<tr>
<td>Moon</td>
<td></td>
<td><anchor>Image Link<go href="moon.wml"/></anchor></td>
</tr>
</table>
</p>
</card>
</wml>
```

Moon, Sun & Earth Page:

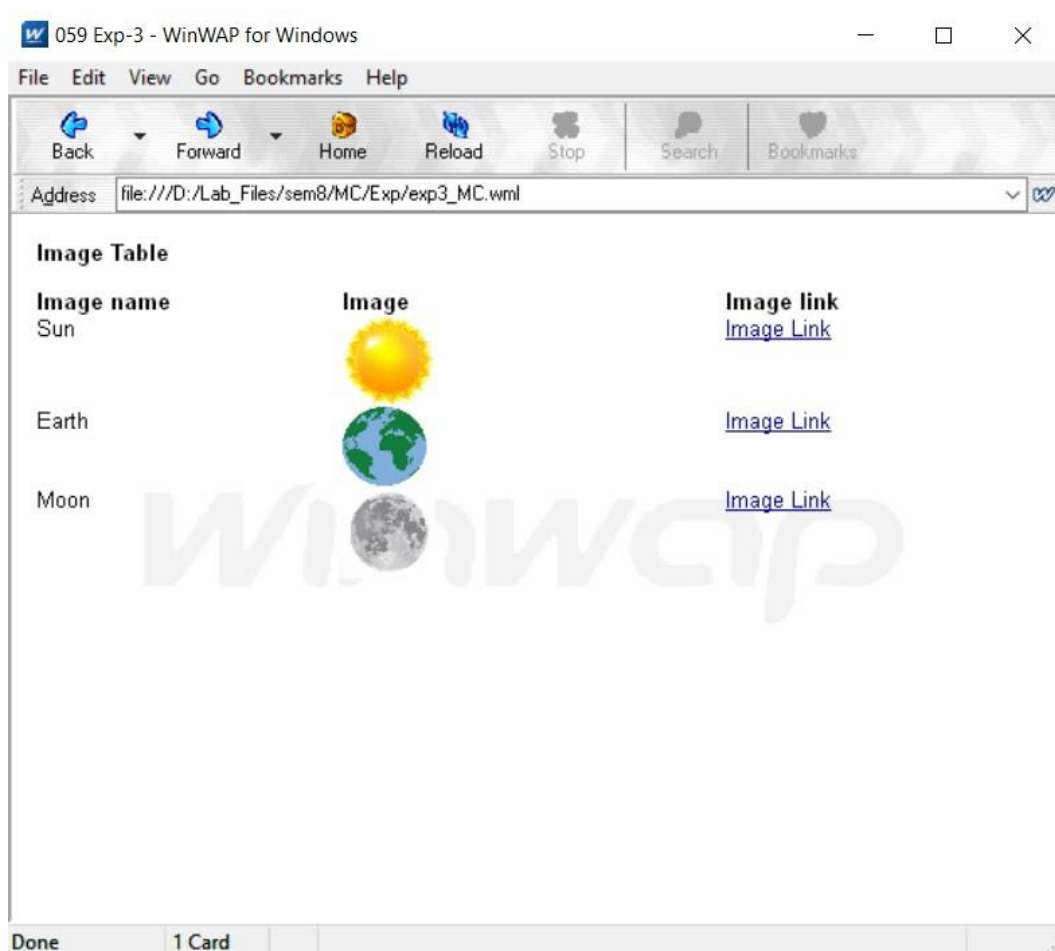
```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card title="059 Exp3 Sun">
<b>Sun</b>
<p></p>
<p>
<anchor> Prev Page
<prev/>
</anchor>
</p>
</card>
</wml>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card title="059 Exp3 Earth">
```

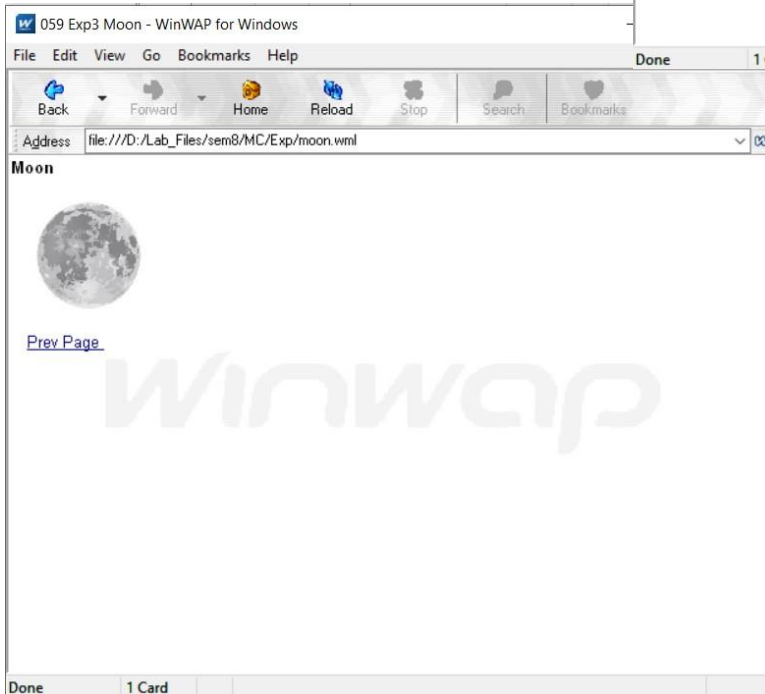
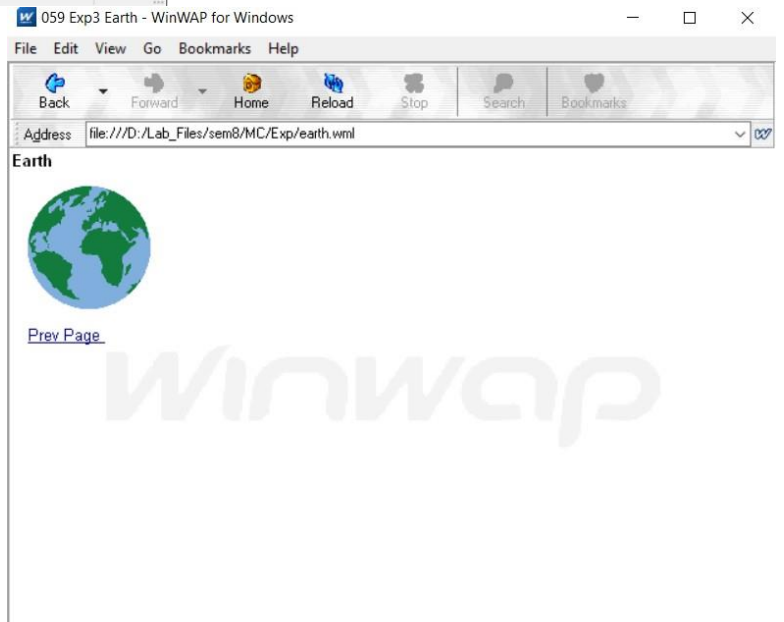
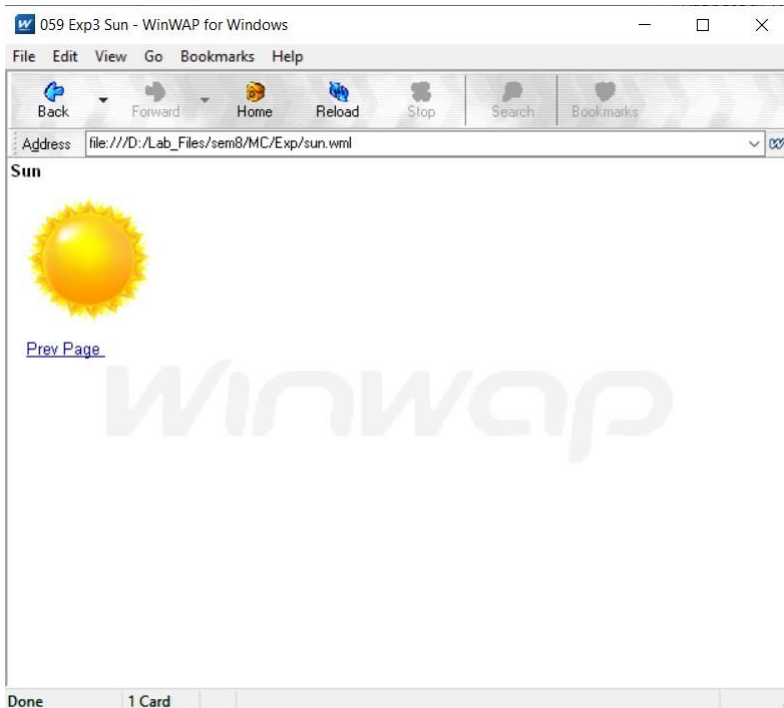
```

<b>Earth</b>
<p></p>
<p>
<anchor> Prev Page
<prev/>
</anchor>
</p>
</card>
</wml>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card title="059 Exp3 Moon">
<b>Moon</b>
<p></p>
<p>
<anchor> Prev Page
<prev/>
</anchor>
</p>
</card>
</wml>

```

## Output:







## **VIVA QUESTIONS**

### **Q1. What is the function of WAP Gateway?**

A WAP gateway sits between mobile devices using the Wireless Application Protocol (WAP) and the World Wide Web, passing pages from one to the other much like a proxy. This translates pages into a form suitable for the mobiles, for instance using the Wireless Markup Language (WML). It decodes the encoded WAP requests from the microbrowser and send the HTTP requests to the internet or to a local application server. It also encodes the WML and HDML data returning from the web for transmission to the microbrowser in the handset.

### **Q2. What is distillation technique in WAP?**

Distillation is a technique to reduce wireless traffic. It is a lossy, real time, data specific Compression.

### **Q3. What is the use of UAProf?**

The UAProf (User Agent Profile) specification is concerned with capturing capability and preference information for wireless devices. This information can be used by content providers to produce content in an appropriate format for the specific device. A UAProf file describes the capabilities of a mobile handset, including Vendor, Model, Screensize, Multimedia Capabilities, Character Set support, and more.

### **Q4. Why WML is called Light Weight Language?**

WML is called a light weight language because it provides programming functionalities but not up to the same extent as Java or C++. With it, the wireless device can do some of the processing and computation.

## Experiment 4

**Aim :** Write a WML program to create a form with multiple options.

**Theory :**

**WML <select> Element:**

The <select>...</select> WML elements are used to define a selection list and the <option>...</option> tags are used to define an item in a selection list. Items are presented as radiobuttons in some WAP browsers. The <option>...</option> tag pair should be enclosed within the <select>...</select> tags.

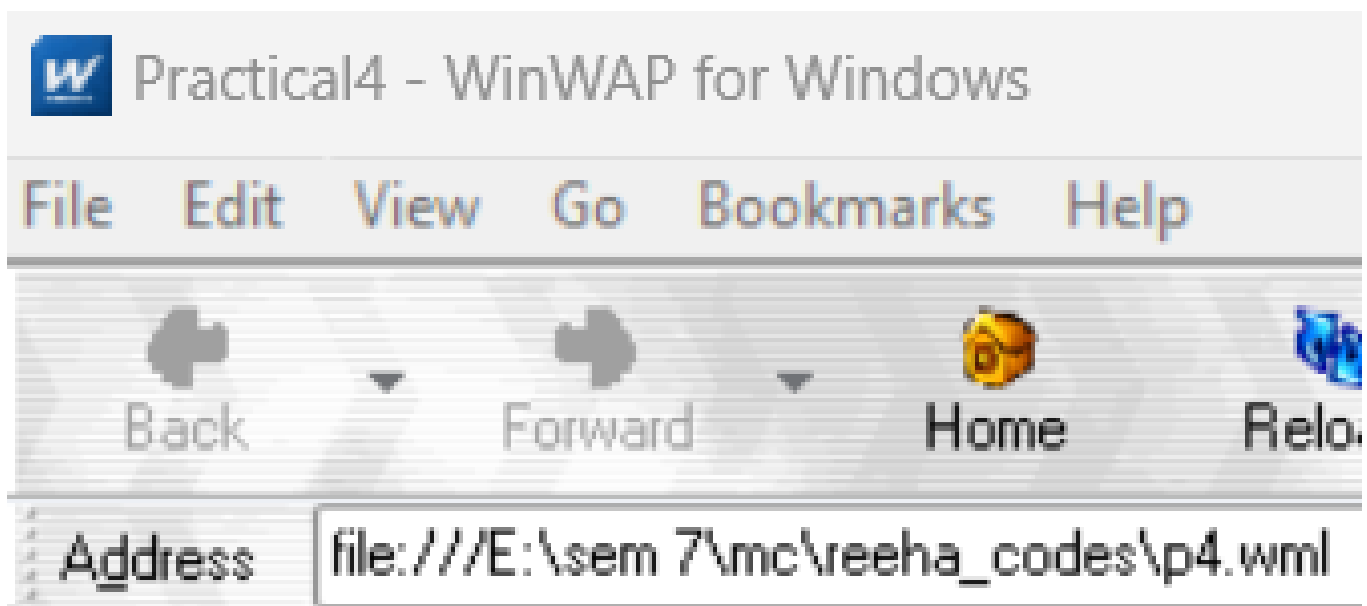
**Attributes:**

Attribute	Value	Description
iname	text	Names the variable that is set with the index result of the selection
ivalue	text	Sets the pre-selected option element
multiple	<ul style="list-style-type: none"><li>true</li><li>false</li></ul>	Sets whether multiple items can be selected. Default is "false"
name	text	Names the variable that is set with the result of the selection
tabindex	number	Sets the tabbing position for the select element
title	text	Sets a title for the list
value	text	Sets the default value of the variable in the "name" attribute
xml:lang	language_code	Sets the language used in the element
class	class data	Sets a class name for the element.
id	element ID	A unique ID for the element.

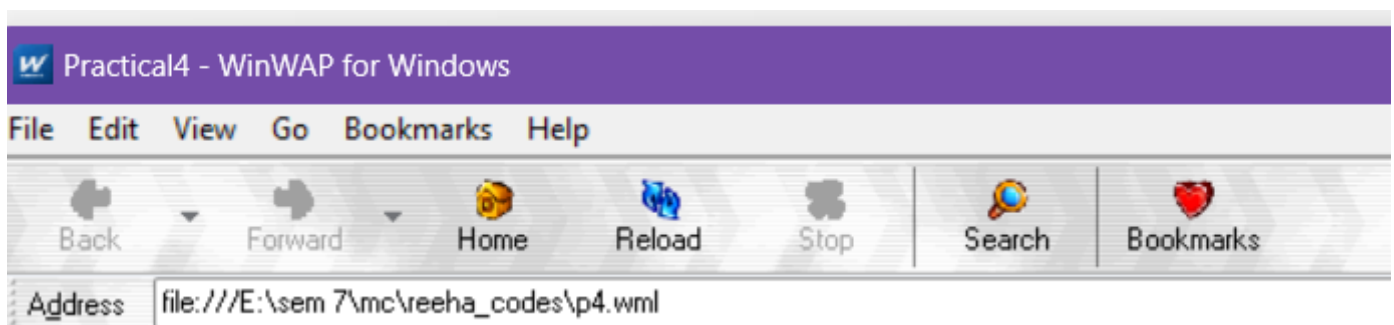
**Code :**

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card title="Practical4">
<p> Select a Tutorial :
<select>
<option value="htm">HTML Tutorial</option>
<option value="xml">XML Tutorial</option>
<option value="wap">WAP Tutorial</option>
</select>
</p>
</card>
</wml>
```

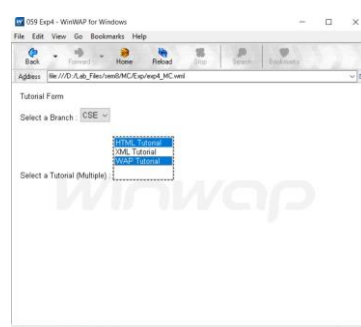
Output :



Select a Tutorial : HTML Tutorial ▾



Select a Tutorial : HTML Tutorial ▾  
HTML Tutorial  
XML Tutorial  
WAP Tutorial



## **VIVA QUESTIONS**

### **Q1. What is Push and Pull technique in WAP?**

In push protocols, the client opens a connection to the server and keeps it constantly active. The server will send (push) all new events to the client using that single always-on connection. In other words, the server PUSHes the new events to the client.

In pull protocols, the client periodically connects to the server, checks for and gets (pulls) recent events and then closes the connection and disconnects from the server. The client repeats this whole procedure to get updated about new events. In this mode, the clients periodically PULLs the new events from the server.

### **Q2. List out the Databases used to store Data of WML pages?**

MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database store data of WML pages.

### **Q3. What is the advantage of using XML DOM document?**

XML DOM is language and platform independent.

XML DOM is traversable - Information in XML DOM is organized in a hierarchy which allows developer to navigate around the hierarchy looking for specific information.

XML DOM is modifiable - It is dynamic in nature providing the developer a scope to add, edit, move or remove nodes at any point on the tree.

### **Q4. What is DTD?**

A DTD is a Document Type Definition. DTD defines the structure and the legal elements and attributes of an XML document. With a DTD, independent groups of people can agree on a standard DTD for interchanging data.

## Experiment 5

**Aim :** Write a WML program to use the time control and to trigger On pick event.

**Theory :**

The ontimer event is used to trigger an event after a given time period. Suppose students wants to display a message after 5 seconds of loading a card then you can use this event to do so.

Here is the syntax to define an event handler for ontimer event:

```
<onevent type="ontimer"> A task to be performed.  
</onevent>
```

```
<timer value="50"/>
```

**Code :**

```
<?xml version="1.0"?>  
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"  
"http://www.wapforum.org/DTD/wml13.dtd">  
<wml>  
<card id="ABC" title="Practical 5">  
<p>
```

Hello World

```
</p>  
<onevent type="ontimer">  
<go href="#ABC1"/>  
</onevent>  
<timer value="100"/>  
</card>  
<card id="ABC1" title="Practical 5">  
<p>Select a Tutorial :  
<select title="tutorials" name="selection_list">  
<option onpick="#xhtml">XHTML Tutorial</option>  
<option onpick="#wap">WAP Tutorial</option>  
</select>  
</p>  
</card>  
<card id="xhtml" title="Practical 5">  
<p>
```

XHTML stands for EXtensible HyperText Markup Language.

```
</p>  
</card>  
<card id="wap" title="Practical 5">  
<p>
```

Wireless Application Protocol (WAP) is a technical standard

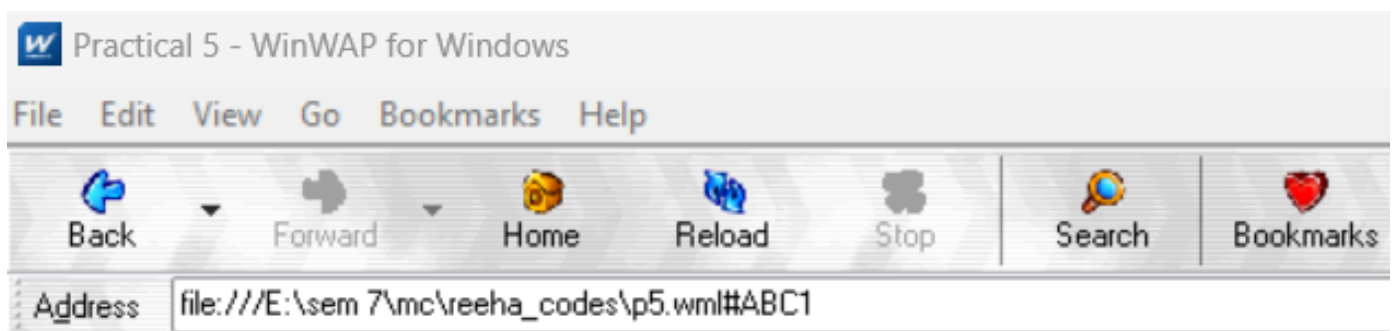
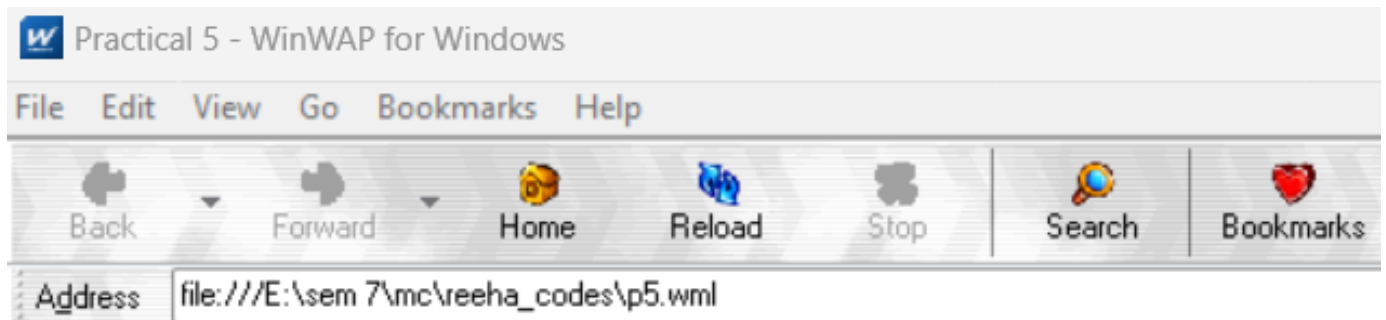
for accessing information over a mobile wireless network.

</p>

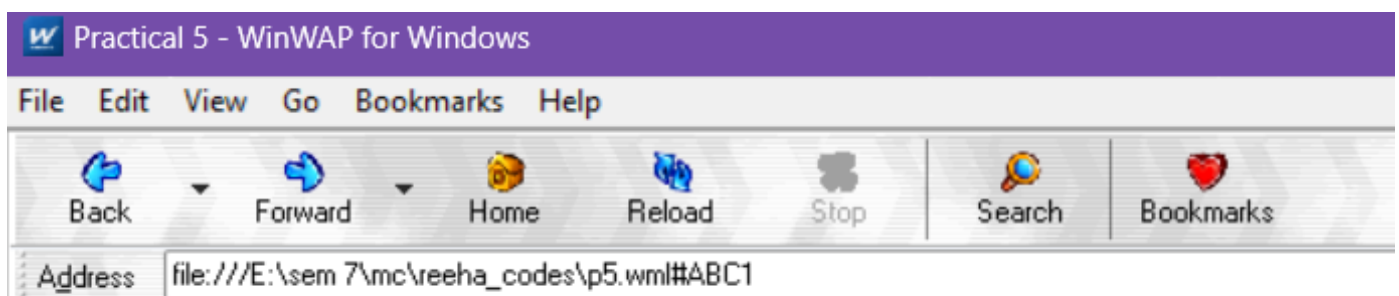
</card>

</wml>

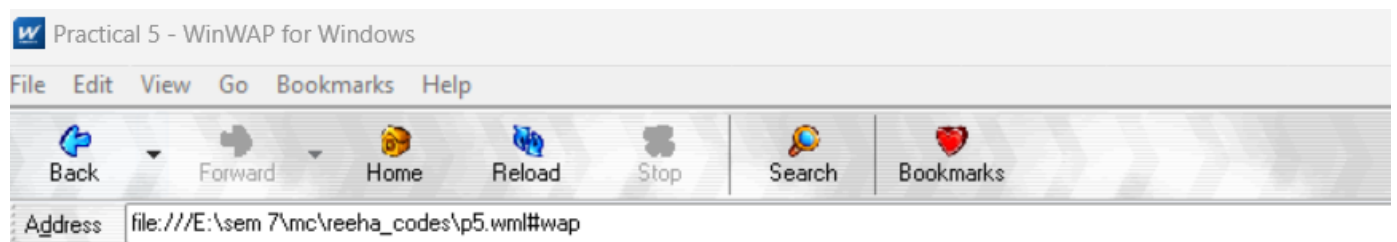
### Output :



Select a Tutorial : XHTML Tutorial ▾

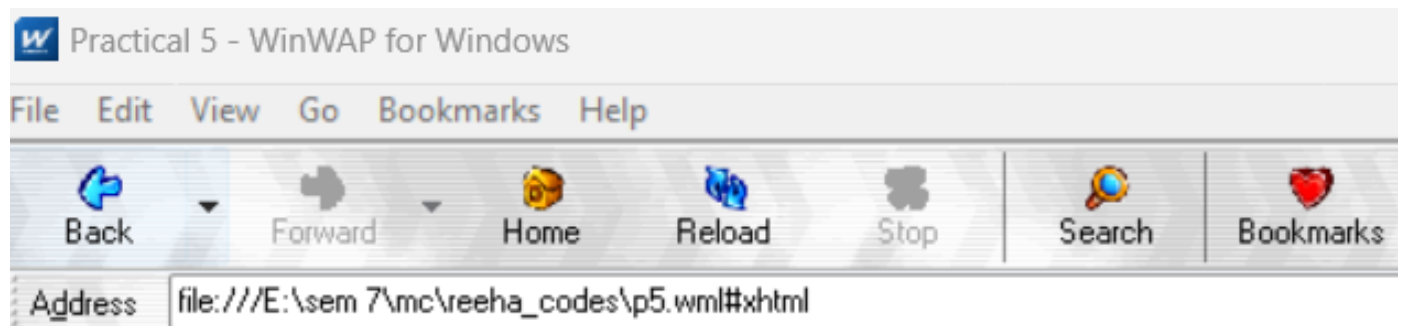


## WAP



Wireless Application Protocol (WAP) is a technical standard for accessing information over a mobile wireless network.

## WML



XHTML stands for EXtensible HyperText Markup Language.

## **VIVA QUESTIONS**

### **Q1. Does WAP run over GPRS?**

Yes, it can do. GPRS is a new over-the-air service that transmits data packets to hand-held devices. It will allow much faster WAP transmission than currently available over SMS or CSD when using GSM.

### **Q2. Which security is used in WAP?**

It has its own security mechanism, named Wireless Transport Layer Security (WTLS). WTLS is a wireless relative of the more common SSL mechanism used by all major web browsers. WTLS resembles SSL in that both rely on certificates on the client and server to verify the identity of the participants involved. While SSL implementations generally rely on RSA encryption, WTLS supports RSA, Diffie-Hellman, and Elliptic Curve encryption. WTLS also doesn't provide for end-to-end security due to WAP's current architecture and limitations of server-side Transport Layer Security (another name for SSL). While WAP clients can securely exchange data with a WAP gateway using WTLS, the gateway must open an SSL session with a back-end server in order to complete the transaction.

### **Q3. Is WML case sensitive?**

WML is case sensitive. No case folding is performed when parsing a WML deck. All enumerated attribute values are case sensitive.

### **Q4. What does Post field tag do?**

The <postfield> tag contains information to be sent to the server along with a <go> tag.



## Experiment 6

**Aim :** Write a WML script to find maximum out of two numbers with help of inbuilt function `Lang.Max()` and to find absolute value with help of inbuilt function `Lang.abs()`

### Theory :

#### Syntax

**Syntax:** `Lang.max(value1, value2)`

#### value1:

The mandatory value1 parameter can be any positive or negative single-precision floating-point number or any positive or negative integer number (including zero). The floating-point number must reside on or between the largest number and the smallest nonzero number supported by the browser. These boundary values can be determined using the `Float.maxFloat` and `Float.minFloat` functions. The integer number must reside on or between the largest positive integer and the largest negative integer supported by the browser. These boundary values can be determined by using the `Lang.maxInt` and `Lang.minInt` functions.

#### value2:

The mandatory value2 parameter can be any positive or negative single-precision floating-point number or any positive or negative integer number (including zero). The floating-point number must reside on or between the largest number and the smallest nonzero number supported by the browser. These boundary values can be determined using the `Float.maxFloat` and `Float.minFloat` functions. The integer number must reside on or between the largest positive integer and the largest negative integer supported by the browser. These boundary values can be determined by using the `Lang.maxInt` and `Lang.minInt` functions.

Returns the maximum of two numbers.

The `Lang.max` function compares two given numbers and then returns the larger value. If the two numbers prove to be equal, the first number is selected. The value and type (floating-point or integer) of the selected number is not changed.

### Code :

#### a) `lang.max()`:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.WAPforum.org/DTD/wml_1.1.xml">
<wml>
<card id="card1">
<p>
max example
</p>
<do type="Accept">
<go href="Max.wmls#findmax()" />
</do>
</card>
<card id="card2">
```

```

<p>
1st number = $(number1)
<br />
2nd number = $(number2)
<br />
maximum number = $(maxnumber)
</p>
</card>
</wml>

```

#### **b) lang.abs():**

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.WAPforum.org/DTD/wml_1.1.xml">
<wml>
<card id="card1">
<p>
abs example
</p>
<do type="accept">
<go href="Abs.wmls#findabs()" />
</do>
</card>
<card id="card2">
<p>
original number = $(number)
<br />
absolute number = $(absnumber)
</p>
</card>
</wml>

```

#### **c) Custom function[findmax()]:**

```

extern function findmax(){
var result1 = Dialogs.prompt("Enter 1st number", "");
var result2 = Dialogs.prompt("Enter 2nd number", "");
var maxnum = Lang.max(result1, result2);
WMLBrowser.setVar("number1", result1);
WMLBrowser.setVar("number2", result2);
WMLBrowser.setVar("maxnumber", maxnum);
WMLBrowser.go("Exp-6.wml#card2");

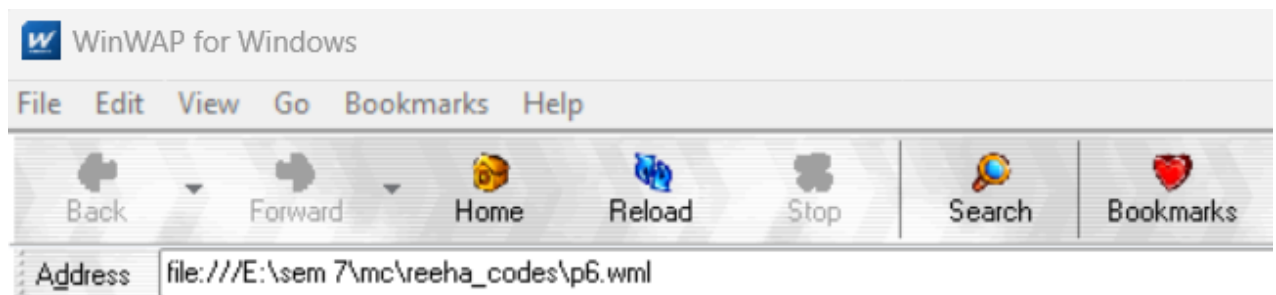
```

```
};
```

```
extern function findabs()  
{  
var result = Dialogs.prompt("Enter any number","");  
var absnum = Lang.abs(result*1);  
WMLBrowser.setVar("number", result);  
WMLBrowser.setVar("absnumber", absnum);  
WMLBrowser.go("Exp-6-b.wml#card2");  
};
```

### Output :

(a)



max example

Accept

Prompt

Enter 1st number

OK Cancel

Prompt

Enter 2nd number

OK Cancel

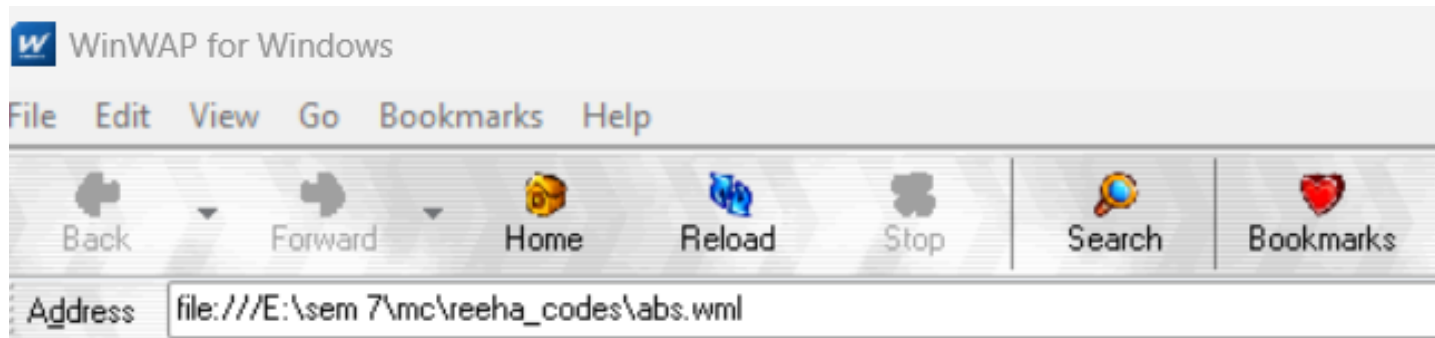
Address file:///F:/Sem 8/MC/Exp-6.wml#card2

1st number = 24

2nd number = 30

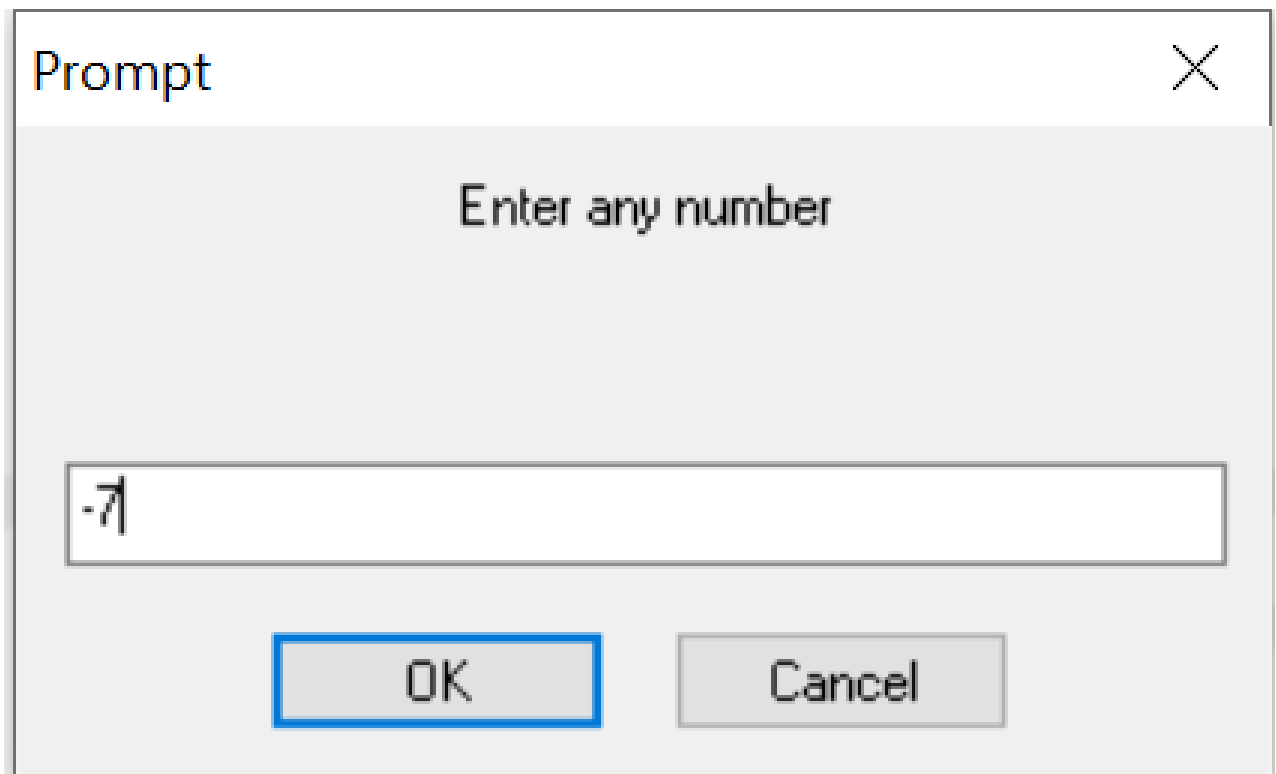
maximum number = 30

b)



abs example

Accept



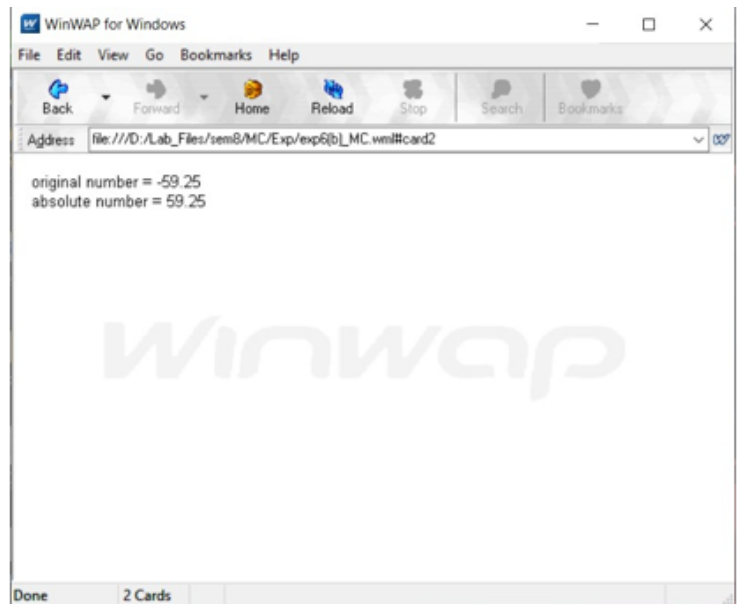
original number = -7  
absolute number = 7

Prompt

Enter any number

-59.25

OK Cancel

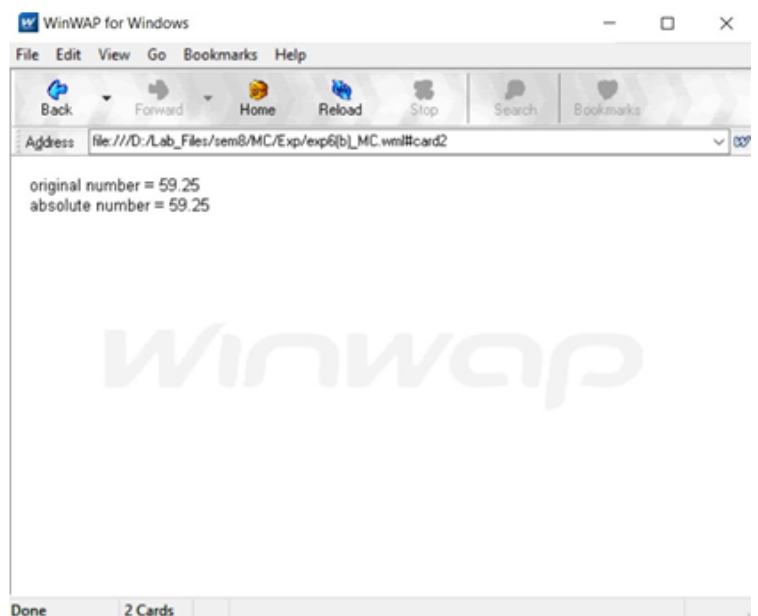


Prompt

Enter any number

59.25

OK Cancel



## VIVA QUESTIONS

### Q1. What are the data types used in WMLScript?

1. Boolean: A Boolean value can be true or false.
2. Integer: WMLScript uses 32-bit integers with two's complement. This means an integer value can be in the range from  $-2^{32}/2$  to  $2^{32}/2-1$ , i.e. -2147483648 to 2147483647.
3. Float: WMLScript uses 32-bit single precision format to represent floating-point numbers. The maximum value supported is  $3.40282347E+38$ . The smallest positive nonzero value supported is  $1.17549435E-38$ .
4. String: A string contains some characters.
5. Invalid. This is used to indicate that a variable is invalid.

### Q2. What are the different components of WMLScript?

- Operators
- Control Statements
- Functions
- Standard Libraries
- Comments
- Case Sensitivity
- White spaces, termination by semi-colon

### Q3. What are the standard libraries used by WMLScripts?

1. Lang – The Lang library provides functions related to the WMLScript language core.
2. Float – The Float library contains functions that help us perform floating-point arithmetic operations.
3. String – The String library provides a number of functions that help us manipulate strings.
4. URL – The URL library contains functions that help us manipulate URLs.
5. WMLBrowser – The WMLBrowser library provides a group of functions to control the WML browser or to get information from it.
6. Dialogs – The Dialogs library Contains the user interface functions.

### Q4. What is the function of WMLScript Control Statements?

Control statements of WMLScript provide a way to control the sequence of the program. They also manage the iterations information used in the program to write the code more efficiently.

## Experiment 7

**Aim :** Write a Program in NS3 to Simulate OLSR.

**Theory :**

**STEPS:**

1. Create a wireless mobile ad-hoc network with three nodes Node1, Node2 and Node3. Install the OLSR routing protocol on these nodes.
2. Place them such that Node1 and Node3 are just out of reach of each other.
3. Create a UDP client on Node1 and the corresponding server on Node3.
4. Schedule Node1 to begin sending packets to Node3 at time 1s.
5. Verify whether Node1 is able to send packets to Node3.
6. Make Node2 move between Node1 and Node3 such that Node2 is visible to both A and C. This should happen at time 20s. Ensure that Node2 stays in that position for another 15s.
7. Verify whether Node1 is able to send packets to Node3.
8. At time 35s, move Node2 out of the region between Node1 and Node3 such that it is out of each other's transmission ranges again.
9. Verify whether Node1 is able to send packets to Node3.
10. To verify whether data transmissions occur in the above scenarios, use either the tracing mechanism or a RecvCallback() for Node3's socket.
11. Plot the number of bytes received versus time at Node3.
12. Show the pcap traces at Node 2's Wifi interface, and indicate the correlation between Node2's packet reception timeline and Node2's mobility.

**Code :**

```
/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 */
```

```

//
// Simple example of OLSR routing over some point-to-point links
//
// Network topology
//
// n0
// \ 5 Mb/s, 2ms
// \      1.5Mb/s, 10ms
// n2 -----n3-----n4
// /
// / 5 Mb/s, 2ms
// n1
//
// - all links are point-to-point links with indicated one-way BW/delay
// - CBR/UDP flows from n0 to n4, and from n3 to n1
// - UDP packet size of 210 bytes, with per-packet interval 0.00375 sec.
// (i.e., DataRate of 448,000 bps)
// - DropTail queues
// - Tracing of queues and packet receptions to file "simple-point-to-point-olsr.tr"

#include <iostream>
#include <fstream>
#include <string>
#include <cassert>

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/olsr-helper.h"
#include "ns3/ipv4-static-routing-helper.h"
#include "ns3/ipv4-list-routing-helper.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE("SimplePointToPointOlsrExample");

int main(int argc, char *argv[])
{
    // Users may find it convenient to turn on explicit debugging
    // for selected modules; the below lines suggest how to do this
    #if 0
    LogComponentEnable ("SimpleGlobalRoutingExample", LOG_LEVEL_INFO);
    #endif

    // Set up some default values for the simulation. Use the

    Config::SetDefault("ns3::OnOffApplication::PacketSize", UIntegerValue(210));
    Config::SetDefault("ns3::OnOffApplication::DataRate", StringValue("448kb/s"));

    // DefaultValue::Bind ("DropTailQueue::m_maxPackets", 30);

    // Allow the user to override any of the defaults and the above

```



```

// DefaultValue::Bind ()s at run-time, via command-line arguments
CommandLine cmd;
cmd.Parse(argc, argv);

// Here, we will explicitly create four nodes. In more sophisticated
// topologies, we could configure a node factory.
NS_LOG_INFO("Create nodes.");
NodeContainer c;
c.Create(5);
NodeContainer n02 = NodeContainer(c.Get(0), c.Get(2));
NodeContainer n12 = NodeContainer(c.Get(1), c.Get(2));
NodeContainer n32 = NodeContainer(c.Get(3), c.Get(2));
NodeContainer n34 = NodeContainer(c.Get(3), c.Get(4));

// Enable OLSR
NS_LOG_INFO("Enabling OLSR Routing.");
OlsrHelper olsr;

Ipv4StaticRoutingHelper staticRouting;

Ipv4ListRoutingHelper list;
list.Add(staticRouting, 0);
list.Add(olsr, 10);

InternetStackHelper internet;
internet.SetRoutingHelper(list); // has effect on the next Install ()
internet.Install(c);

// We create the channels first without any IP addressing information
NS_LOG_INFO("Create channels.");
PointToPointHelper p2p;
p2p.SetDeviceAttribute("DataRate", StringValue("5Mbps"));
p2p.SetChannelAttribute("Delay", StringValue("2ms"));
NetDeviceContainer nd02 = p2p.Install(n02);
NetDeviceContainer nd12 = p2p.Install(n12);
p2p.SetDeviceAttribute("DataRate", StringValue("1500kbps"));
p2p.SetChannelAttribute("Delay", StringValue("10ms"));
NetDeviceContainer nd32 = p2p.Install(n32);
NetDeviceContainer nd34 = p2p.Install(n34);

// Later, we add IP addresses.
NS_LOG_INFO("Assign IP Addresses.");
Ipv4AddressHelper ipv4;
ipv4.SetBase("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i02 = ipv4.Assign(nd02);

ipv4.SetBase("10.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer i12 = ipv4.Assign(nd12);

ipv4.SetBase("10.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer i32 = ipv4.Assign(nd32);

ipv4.SetBase("10.1.4.0", "255.255.255.0");
Ipv4InterfaceContainer i34 = ipv4.Assign(nd34);

```

```

// Create the OnOff application to send UDP datagrams of size
// 210 bytes at a rate of 448 Kb/s from n0 to n4
NS_LOG_INFO("Create Applications.");
uint16_t port = 9; // Discard port (RFC 863)

OnOffHelper onoff("ns3::UdpSocketFactory",
                  InetSocketAddress(i34.GetAddress(1), port));
onoff.SetConstantRate(DataRate("448kb/s"));

ApplicationContainer apps = onoff.Install(c.Get(0));
apps.Start(Seconds(1.0));
apps.Stop(Seconds(10.0));

// Create a packet sink to receive these packets
PacketSinkHelper sink("ns3::UdpSocketFactory",
                      InetSocketAddress(Ipv4Address::GetAny(), port));

apps = sink.Install(c.Get(3));
apps.Start(Seconds(1.0));
apps.Stop(Seconds(10.0));

// Create a similar flow from n3 to n1, starting at time 1.1 seconds
onoff.SetAttribute("Remote",
                  AddressValue(InetSocketAddress(i12.GetAddress(0), port)));
apps = onoff.Install(c.Get(3));
apps.Start(Seconds(1.1));
apps.Stop(Seconds(10.0));

// Create a packet sink to receive these packets
apps = sink.Install(c.Get(1));
apps.Start(Seconds(1.1));
apps.Stop(Seconds(10.0));

AsciiTraceHelper ascii;
p2p.EnableAsciiAll(ascii.CreateFileStream("simple-point-to-point-olsr.tr"));
p2p.EnablePcapAll("simple-point-to-point-olsr");

Simulator::Stop(Seconds(30));

NS_LOG_INFO("Run Simulation.");
Simulator::Run();
Simulator::Destroy();
NS_LOG_INFO("Done.");

return 0;
}

```

## Output :

```
● reeha@Reeha:~/networkEng/ns-allinone-3.35/ns-3.35$ ./waf --run exp7
Waf: Entering directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
[2958/3007] Compiling scratch/exp7.cc
[2968/3007] Linking build/scratch/exp7
Waf: Leaving directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (5.119s)
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:0 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:1 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:2 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:3 Does not have a mobility model. Use SetConstantPosition if it is stationary
AnimationInterface WARNING:Node:4 Does not have a mobility model. Use SetConstantPosition if it is stationary
● reeha@Reeha:~/networkEng/ns-allinone-3.35/ns-3.35$ ./waf --run exp7
Waf: Entering directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
[2958/3007] Compiling scratch/exp7.cc
[2968/3007] Linking build/scratch/exp7
Waf: Leaving directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3.930s)

● reeha@Reeha:~/networkEng/ns-allinone-3.35/ns-3.35$ ./waf --run exp7
Waf: Entering directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
[2945/3007] Compiling scratch/exp7.cc
[2968/3007] Linking build/scratch/exp7
Waf: Leaving directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.033s)
○ reeha@Reeha:~/networkEng/ns-allinone-3.35/ns-3.35$ █
```

## **VIVA QUESTIONS**

### **Q1. List of Security Issues in Adhoc Networks?**

The constantly changing nature of the network topology coupled with data transmission in open medium makes it highly susceptible to attacks. Security issues with respect to data confidentiality, availability of systems and applications, authentication, system integrity are just as threatening as in conventional networks. Vulnerabilities can lead to message eavesdropping, injection of fake messages, denial of service attack or poor monitoring of routing information.

MANETs are susceptible to both internal and external attacks.

### **Q2. What is Multi Casting?**

Multicasting in computer network is a group communication, where a sender(s) send data to multiple receivers simultaneously. It supports one – to – many and many – to – many data transmission across LANs or WANs. Through the process of multicasting, the communication and processing overhead of sending the same data packet or data frame is minimized.

### **Q3. What is MANET?**

MANET stands for Mobile adhoc Network also called as wireless adhoc network or adhoc wireless network that usually has a routable networking environment on top of a Link Layer ad hoc network. They consist of set of mobile nodes connected wirelessly in a self-configured, self-healing network without having a fixed infrastructure. MANET nodes are free to move randomly as the network topology changes frequently. Each node behaves as a router as they forward traffic to other specified node in the network.

### **Q4. What are the Characteristics of MANETs?**

- Dynamic topologies
- Bandwidth constrained, variable capacity links
- Autonomous behavior
- Energy constrained operation
- Limited security
- Less human intervention

## Experiment 8

**Aim :** Write a Program in NS3 to Simulate AODV

### Theory :

Ad hoc On-Demand Distance Vector (AODV) [1] is a routing protocol for mobile ad-hoc networks. AODV is a reactive routing protocol which means that routes are discovered when necessary, routes are also maintained as long as necessary. Every node maintains its monotonically increasing sequence number which increases every time the node notices change in the neighborhood topology. Routing tables are used to store routing information where data is stored as follows:

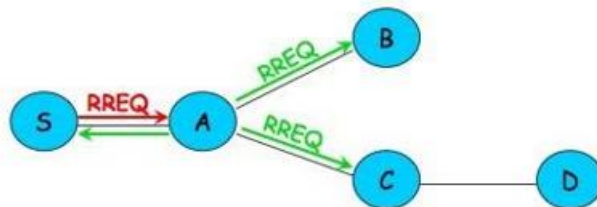
**<destination addr, next-hop addr, destination sequence number, life\_time>**

life\_time gets updated every time the route is used (If not used within its life\_time, route expires).

### AODV Routing Protocol

AODV Routing Protocol works as follows: Assuming in the network in the figure below we need to discover route between S and D.

Figure 1: AODV protocol working on five nodes.

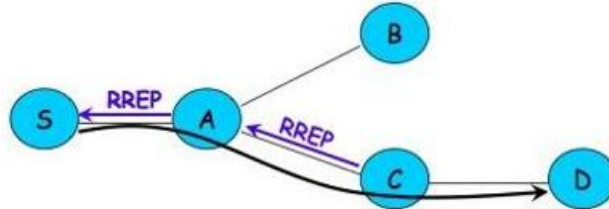


It works as follows:

1. Node S needs route to D (no routes in routing table)
2. S Creates a Route Request (RREQ): D's IP addr, D's seq #, S's IP addr, S's seq #, hopcount (0)
3. Node S broadcasts RREQ to neighbors
4. Node A receives RREQ
  - Makes reverse route entry for S (dest=S, nexthop=S, hopcount=1)
  - (b) It has no route to D, so rebroadcasts RREQ
5. Node C receives RREQ
  - Makes reverse route entry for S (dest=S, nexthop=A, hopcount=2)
  - It has a route to D, and the seq # for route to D is  $\geq$  D's seq # in RREQ (c) C creates a Route Reply (RREP)
  - Enter D's IPAddr, seq#, S's IPAddr, Hopcount to D (1)
  - Unicasts RREP to A
6. Node A receives RREP
  - a. Makes forward route entry to D (dest=D, nexthop=C,

- hopcount=2) (b) Unicasts RREP to S
7. Node S receives RREP
    - a. Makes forward route entry to D (dest = D, nexthop = A, hopcount=3) (b) Sends data packets on route to D

Figure 2: AODV protocol: Node S receives PREP.



### Code :

```

/* -*- Mode:C++; c-file-style:"gnu"; indent-tabs-mode:nil; -*- */
/*
 * Copyright (c) 2009 IITP RAS
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation;
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software
 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
 *
 * This is an example script for AODV manet routing protocol.
 *
 * Authors: Pavel Boyko <boyko@iitp.ru>
 */

#include <iostream>
#include <cmath>
#include "ns3/aodv-module.h"
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/mobility-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/v4ping-helper.h"
#include "ns3/yans-wifi-helper.h"

using namespace ns3;

class AodvExample

```

```

{
public:
    AodvExample ();
    bool Configure (int argc, char **argv);
    void Run ();
    void Report (std::ostream & os);

private:

    // parameters
    uint32_t size;
    double step;
    double totalTime;
    bool pcap;
    bool printRoutes;

    // network
    NodeContainer nodes;
    NetDeviceContainer devices;
    Ipv4InterfaceContainer interfaces;

private:
    void CreateNodes ();
    void CreateDevices ();
    void InstallInternetStack ();
    void InstallApplications ();
};

int main (int argc, char **argv)
{
    AodvExample test;
    if (!test.Configure (argc, argv))
        NS_FATAL_ERROR ("Configuration failed. Aborted.");

    test.Run ();
    test.Report (std::cout);
    return 0;
}

//-----
AodvExample::AodvExample () :
    size (10),
    step (100),
    totalTime (100),
    pcap (true),
    printRoutes (true)
{
}

bool
AodvExample::Configure (int argc, char **argv)
{
    // Enable AODV logs by default. Comment this if too noisy
    // LogComponentEnable("AodvRoutingProtocol", LOG_LEVEL_ALL);

```

[illegible]



```

        "LayoutType", StringValue ("RowFirst"));
    mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
    mobility.Install (nodes);
}

void
AodvExample::CreateDevices ()
{
    WifiMacHelper wifiMac;
    wifiMac.SetType ("ns3::AdhocWifiMac");
    YansWifiPhyHelper wifiPhy = YansWifiPhyHelper();
    YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::Default ();
    wifiPhy.SetChannel (wifiChannel.Create ());
    WifiHelper wifi;
    wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode",
    StringValue ("OfdmRate6Mbps"), "RtsCtsThreshold", UIntegerValue (0));
    devices = wifi.Install (wifiPhy, wifiMac, nodes);

    if (pcap)
    {
        wifiPhy.EnablePcapAll (std::string ("aodv"));
    }
}

void
AodvExample::InstallInternetStack ()
{
    AodvHelper aodv;
    // you can configure AODV attributes here using aodv.Set(name, value)
    InternetStackHelper stack;
    stack.SetRoutingHelper (aodv); // has effect on the next Install ()
    stack.Install (nodes);
    Ipv4AddressHelper address;
    address.SetBase ("10.0.0.0", "255.0.0.0");
    interfaces = address.Assign (devices);

    if (printRoutes)
    {
        Ptr<OutputStreamWrapper> routingStream = Create<OutputStreamWrapper>
("aodv.routes", std::ios::out);
        aodv.PrintRoutingTableAllAt (Seconds (8), routingStream);
    }
}

void
AodvExample::InstallApplications ()
{
    V4PingHelper ping (interfaces.GetAddress (size - 1));
    ping.SetAttribute ("Verbose", BooleanValue (true));

    ApplicationContainer p = ping.Install (nodes.Get (0));
    p.Start (Seconds (0));
    p.Stop (Seconds (totalTime) - Seconds (0.001));

    // move node away

```

```

Ptr<Node> node = nodes.Get (size/2);
Ptr<MobilityModel> mob = node->GetObject<MobilityModel> ();
Simulator::Schedule (Seconds (totalTime/3), &MobilityModel::SetPosition, mob, Vector
(1e5, 1e5, 1e5));
}

```

## Output :

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

Waf: Entering directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Waf: Leaving directory `/home/reeha/networkEng/ns-allinone-3.35/ns-3.35/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.011s)
Creating 10 nodes 50 m apart.
Starting simulation for 100 s ...
PING 10.0.0.10 - 56 bytes of data - 84 bytes including ICMP and IPv4 headers.
64 bytes from 10.0.0.10: icmp_seq=0 ttl=56 time=+2057.55ms
64 bytes from 10.0.0.10: icmp_seq=1 ttl=56 time=+1059.66ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=56 time=+61.0477ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=56 time=+7.36502ms
64 bytes from 10.0.0.10: icmp_seq=4 ttl=56 time=+7.28402ms
64 bytes from 10.0.0.10: icmp_seq=5 ttl=56 time=+7.26602ms
64 bytes from 10.0.0.10: icmp_seq=6 ttl=56 time=+7.36502ms
64 bytes from 10.0.0.10: icmp_seq=7 ttl=56 time=+7.31102ms
64 bytes from 10.0.0.10: icmp_seq=8 ttl=56 time=+7.34702ms
64 bytes from 10.0.0.10: icmp_seq=9 ttl=56 time=+7.38302ms
64 bytes from 10.0.0.10: icmp_seq=10 ttl=56 time=+7.35602ms
64 bytes from 10.0.0.10: icmp_seq=11 ttl=56 time=+7.31102ms
64 bytes from 10.0.0.10: icmp_seq=12 ttl=56 time=+7.35602ms
64 bytes from 10.0.0.10: icmp_seq=13 ttl=56 time=+7.30202ms
64 bytes from 10.0.0.10: icmp_seq=14 ttl=56 time=+7.25702ms
64 bytes from 10.0.0.10: icmp_seq=15 ttl=56 time=+7.30202ms
64 bytes from 10.0.0.10: icmp_seq=16 ttl=56 time=+7.32002ms
64 bytes from 10.0.0.10: icmp_seq=17 ttl=56 time=+7.32902ms
64 bytes from 10.0.0.10: icmp_seq=18 ttl=56 time=+7.34702ms
64 bytes from 10.0.0.10: icmp_seq=19 ttl=56 time=+7.37402ms
64 bytes from 10.0.0.10: icmp_seq=20 ttl=56 time=+7.30202ms
64 bytes from 10.0.0.10: icmp_seq=21 ttl=56 time=+7.27502ms
64 bytes from 10.0.0.10: icmp_seq=22 ttl=56 time=+7.28402ms
64 bytes from 10.0.0.10: icmp_seq=23 ttl=56 time=+7.27502ms
64 bytes from 10.0.0.10: icmp_seq=24 ttl=56 time=+7.36502ms
64 bytes from 10.0.0.10: icmp_seq=25 ttl=56 time=+7.32002ms
64 bytes from 10.0.0.10: icmp_seq=26 ttl=56 time=+7.29302ms
64 bytes from 10.0.0.10: icmp_seq=27 ttl=56 time=+7.29302ms
64 bytes from 10.0.0.10: icmp_seq=28 ttl=56 time=+7.32902ms
64 bytes from 10.0.0.10: icmp_seq=29 ttl=56 time=+7.33802ms
64 bytes from 10.0.0.10: icmp_seq=30 ttl=56 time=+7.27502ms
64 bytes from 10.0.0.10: icmp_seq=31 ttl=56 time=+7.32902ms
64 bytes from 10.0.0.10: icmp_seq=32 ttl=56 time=+7.31102ms
64 bytes from 10.0.0.10: icmp_seq=33 ttl=56 time=+7.38302ms
--- 10.0.0.10 ping statistics ---
100 packets transmitted, 34 received, 66% packet loss, time +1e+05ms
rtt min/avg/max/mdev = 7/99.82/2057/390 ms
reeha@Reeha:~/networkEng/ns-allinone-3.35/ns-3.35$

```

## **VIVA QUESTIONS**

### **Q1. How routing in Adhoc networks different from fixed networks?**

In infrastructure mode, the routing part is handled by the access point and the distribution system; every wireless device just need to forward all its data packets to this access point. But, in AD Hoc networks, there is no common access point for connections, and, every device acts as a router. This scenario is totally new. Adding to this, devices are not fixed, they can be mobile, contrary to the Internet where every router has fixed neighbours (excepts if a link fails).

### **Q2. What is hidden and exposed terminal problem in Adhoc Networks?**

In wireless networks, the exposed node problem occurs when a node is prevented from sending packets to other nodes because of co-channel interference with a neighboring transmitter.

In wireless networking, the hidden node problem or hidden terminal problem occurs when a node can communicate with a wireless access point (AP), but cannot directly communicate with other nodes that are communicating with that AP.

### **Q3. What is Hiper Access?**

HiperAccess is an ETSI (European Telecommunications Standards Institute), now 3GPP (Third Generation Partnership Project), standard used to provide outdoor, high speed (25Mbps typical data rate) fixed radio access to customer premises. The system is capable of supporting multi-media applications and will be operated in either licensed or licensed exempted spectrum – typically 5GHz.

### **Q4. What are hybrid routing Protocols?**

Hybrid protocols utilize the capabilities of both reactive and proactive protocols, and unite them together to achieve better results. The network is separated into zones and use different protocols in two different zones that one protocol is used within the zone and other protocol is used between them. An example of hybrid routing protocol is zone routing protocol (ZRP).

## Experiment 9

**Aim :** Write an Android Program to create list view, grid view and database connectivity.

**Code :**

### LIST VIEW -----

#### MainActivity.java

```
package com.example.listview;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    ListView listView;

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

        listView = (ListView) findViewById(R.id.listview);

        ArrayList<String> arrayList = new ArrayList<>();
        arrayList.add("Java");
        arrayList.add("C++");
        arrayList.add("Python");
        arrayList.add("JavaScript");
        arrayList.add("C");

        ArrayAdapter arrayAdapter = new ArrayAdapter(this,
        android.R.layout.simple_list_item_1, arrayList);

        listView.setAdapter(arrayAdapter);
    }
}
```

#### activity\_main.xml

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

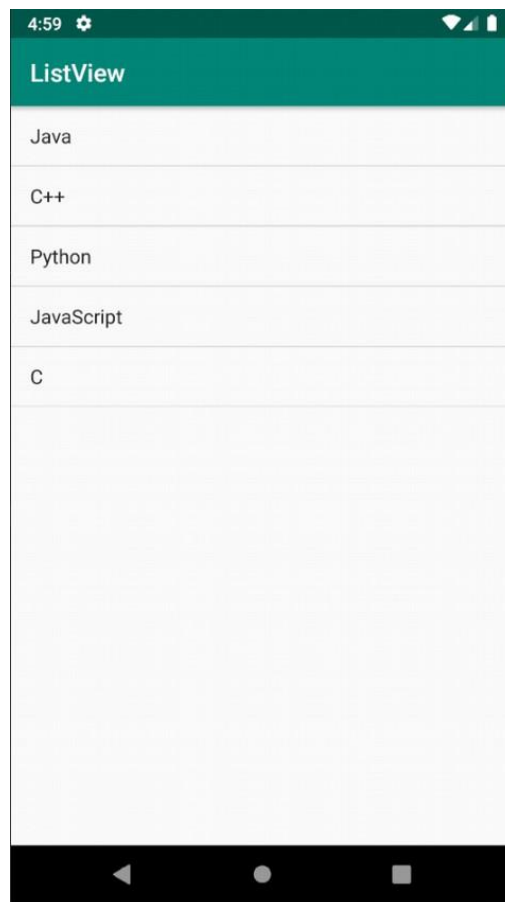
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"  
android:layout_height="match_parent"  
tools:context=".MainActivity">
```

```
<ListView  
android:id="@+id/listview" android:layout_width="match_parent"  
android:layout_height="match_parent" />
```

```
</RelativeLayout>
```

## Output:



## GRID VIEW -----

### MainAcctivity.java

```
package com.example.gridview;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;
```

```

import android.widget.GridView;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    GridView gridView;
    static final String[] gridViewValue = new String[] { "FOC", "DSA", "CN",
        "OOPs", "STQA", "COA",
        "ADBMS", "DBMS", "OS",
        "SE", "TOC", "WebD",
        "MC", "PPL", "SC"
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        gridView = findViewById(R.id.gridView);

        ArrayAdapter adapter = new ArrayAdapter(this, android.R.layout.simple_list_item_1,
        gridViewValue);

        gridView.setAdapter(adapter);

    }
}

```

### **activity\_main.xml**

```

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

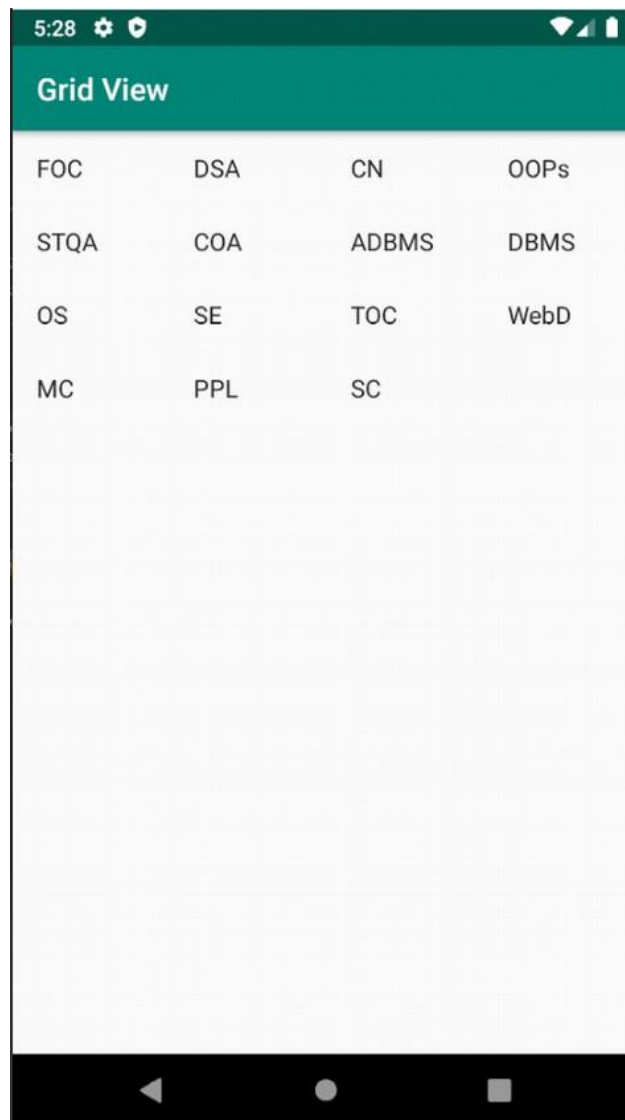
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <GridView
        android:layout_width="match_parent" android:layout_height="match_parent"
        android:id="@+id/gridView" android:columnWidth="100dp" android:minHeight="100dp"
        android:numColumns="auto_fit" android:stretchMode="columnWidth"/>

</RelativeLayout>

```

## Output:



FOC	DSA	CN	OOPs
STQA	COA	ADBMS	DBMS
OS	SE	TOC	WebD
MC	PPL	SC	

## DATABASE CONNECTIVITY -----

### MainActivity.java

```
import android.widget.Toast;

import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;

public class MainActivity extends AppCompatActivity {

    // Initializing fields DatabaseHelper myDB;
    EditText edit_name, edit_surname, edit_marks, edit_id;
    Button addData, viewData, updateData, deleteData;
    String name, surname, marks, id;
    boolean isUpdated;
    TextView textView;
```

```

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

    //Initialize Database
    myDB = new DatabaseHelper( this );

    //Initialize EditText
    edit_name = findViewById( R.id.name );
    edit_surname = findViewById( R.id.surname );
    edit_marks = findViewById( R.id.marks );
    edit_id = findViewById( R.id.id );

    // TextView
    textView = findViewById( R.id.textView5 );
    textView.setText( "Important Notes:\n1. Both Date and Time will be stored automatically on
the time of insertion.\n2. Existing Date and Time will be updated when you update your data."
);

    //Initialize Button
    addData = findViewById( R.id.button );
    viewData = findViewById( R.id.button2 );
    updateData = findViewById( R.id.button3 );
    deleteData = findViewById( R.id.button4 );

    //Call Methods AddData(); viewData(); updateData(); deleteData();
}

//Adding or inserting data to database public void AddData(){

addData.setOnClickListener( new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        name = edit_name.getText().toString();
        surname = edit_surname.getText().toString();
        marks = edit_marks.getText().toString();

        //Current Date and Time Date date1 = new Date();
        String date = DateFormat.getDateTimeInstance().format(date1);
        boolean isInserted = myDB.instertData( name, surname, marks, date);

        if(isInserted == true){
            Toast.makeText( MainActivity.this, "Data is inserted", Toast.LENGTH_SHORT ).show();
        }
        else
            Toast.makeText( MainActivity.this, "Data is not inserted", Toast.LENGTH_SHORT
        ).show();
    }
});
}

//For viewing data in database public void viewData(){

```



```
viewData.setOnClickListener(
    new View.OnClickListener() {

        @Override
public void onClick(View view)
{
    Cursor res = myDB.getData();

if (res.getCount() == 0)
{
    showMessage("Error", "Data not found!");
}

else{
StringBuffer buffer = new StringBuffer();
while (res.moveToNext())
{
buffer.append( "ID: " + res.getString( 0 ) + "\n" );
buffer.append( "Name: " + res.getString( 1 ) + "\n" );
buffer.append( "Surname: " + res.getString( 2 ) + "\n" );
buffer.append( "Marks: " + res.getString( 3 ) + "\n" );
buffer.append( "Insertion/Udation Date:\n" + res.getString( 4 ) + "\n\n" );
}

showMessage( "Data", buffer.toString() );


//For updating existing data in database public void updateData(){

updateData.setOnClickListener( new View.OnClickListener()
{
    @Override
public void onClick(View view) {

id = edit_id.getText().toString();
name = edit_name.getText().toString();
surname = edit_surname.getText().toString();
marks = edit_marks.getText().toString();


//Current Date and Time Date date1 = new Date();
String date = DateFormat.getDateInstance(). format(date1);


boolean isUpdated = myDB.updateData( id, name, surname, marks, date);



if (isUpdated == true){
showMessage( "Update", "Your data has been successfully updated!" );
} else {
showMessage( "Update failed", "Cannot Update your data :(" );}




//For deleting data in the database public void deleteData(){

deleteData.setOnClickListener( new View.OnClickListener() { @Override
public void onClick(View view) {

id = edit_id.getText().toString();
```

```

        Integer res = myDB.deleteData( id ); if(res > 0){
            Toast.makeText( getApplicationContext(), "Row effected", Toast.LENGTH_SHORT
        ).show();
        }
        else{
            Toast.makeText( getApplicationContext(), "Row not effected", Toast.LENGTH_SHORT
        ).show();}
        } );
    }

    // Method for creating AlertDialog box
    private void showMessage(String title, String message) {

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }
}

```

### **activity\_main.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="com.example.laksh.sqlapp.MainActivity">

    <Button
        android:id="@+id/button" android:layout_width="wrap_content"
        android:layout_height="52dp" android:text="Add Data"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.054"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.84" />

    <TextView
        android:id="@+id/textView" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:text="Name"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.049" app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" app:layout_constraintVertical_bias="0.063" />

    <EditText
        android:id="@+id/name" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:ems="10"
        android:inputType="textPersonName" app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.502"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.034" />

```

```
<TextView
android:id="@+id/textView2" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Surname"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.049"
app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.17" />
```

```
<EditText
android:id="@+id/surname" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:ems="10"
android:inputType="textPersonName" app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.502"
app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.133" />
```

```
<TextView
android:id="@+id/textView3" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Marsks"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.049"
app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.258" />
```

```
<EditText
android:id="@+id/marks" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:ems="10"
android:inputType="textPersonName" app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.502"
app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.232" />
```

```
<Button
android:id="@+id/button2" android:layout_width="wrap_content"
android:layout_height="51dp" android:text="View Data"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.447"
app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.84" />
```

```
<Button
android:id="@+id/button3" android:layout_width="wrap_content"
android:layout_height="51dp" android:text="Update data"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.939"
app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.84" />
```

```
<EditText
android:id="@+id/id" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:ems="10"
android:inputType="textPersonName" app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.5"
app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
```

```

app:layout_constraintVertical_bias="0.331" />

<Button
    android:id="@+id/button4" android:layout_width="wrap_content"
    android:layout_height="50dp" android:text="delete data"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.474"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.984" />

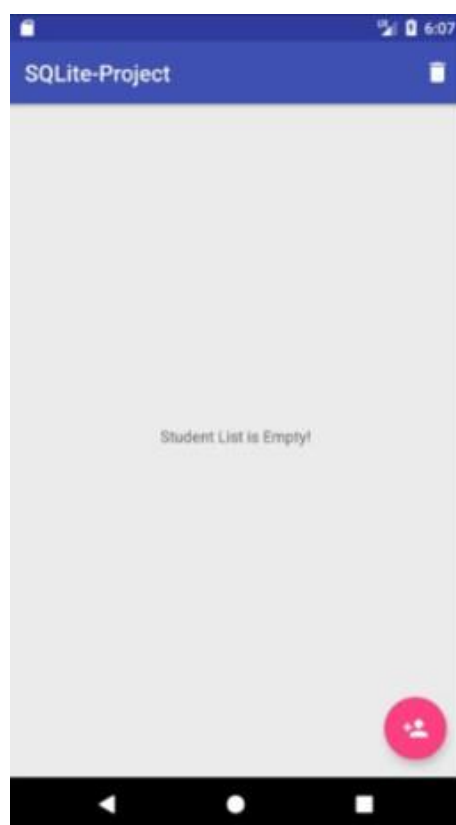
<TextView
    android:id="@+id/textView4" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="Id"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.049"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.339" />

<TextView
    android:id="@+id/textView5" android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:text="TextView" android:paddingStart="10dp"
    android:paddingEnd="10dp"
    android:layout_marginTop="20dp" app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.049"
    app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.453" />

</android.support.constraint.ConstraintLayout>

```

## Output:





## **VIVA QUESTIONS**

### **Q1. What is View group in Android?**

A ViewGroup is a special view that can contain other views. The ViewGroup is the base class for Layouts in android, like Linear Layout, Relative Layout , Frame Layout etc. In other words, ViewGroup is generally used to define the layout in which views(widgets) will be set/arranged/listed on the android screen.

### **Q2. What is a Content Provider in Android?**

A content provider manages access to a central repository of data. A provider is part of an Android application, which often provides its own UI for working with the data. However, content providers are primarily intended to be used by other applications, which access the provider using a provider client object.

### **Q3. What is Container in Android?**

A container is a view used to contain other views. Android offers a collection of view classes that act as containers for views. These container classes are called layouts, and as the name suggests, they decide the organization, size, and position of their children views.

### **Q4. What is a ADB and ANR in Android?**

The Android Debug Bridge (ADB) command line utility lets you communicate with an instance of an Android emulator or an Android device that is connected.

ANR or Application Not Responding is an android error that happens when the UI thread becomes unresponsive to the user. When the application becomes unresponsive, the user is presented with an ANR dialog that gives them the option to force quit the application.

**You can access ANR traces from a device or emulator by using Android Debug Bridge (adb).**

## Experiment 9

**Aim :** Make an application of your choice using WML or Android.

**Code :**

Phone Book using WML Script.

**Phonebook.wml File:**

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
  <card id="card1" title="Phone Book" newcontext="true">
    <p>
      <b>Phone Book</b>
    </p>

    <p>
      Select Contact Name:
      <select name="name" value="Reeha" title="Name">
        <option value="Reeha">Reeha</option>
        <option value="Saba">Saba</option>
        <option value="Shruti">Shruti</option>
        <option value="Saleena">Saleena</option>
      </select>
      <br><br>
      <do type="accept" label="Get number">
        <go href="Phonebook.wmls#phone()"/>
      </do>
    </p>
  </card>

  <card id="card2" title="Phone Number">
    <p>
      Contact Details
    </p>
    <p>
      <b>$(fullname)</b><br>
      <b>$(phoneNumber)</b><br>
      <b>$(email)</b>
    </p>
  </card>
</wml>
```

## Phonebook.wmls File:

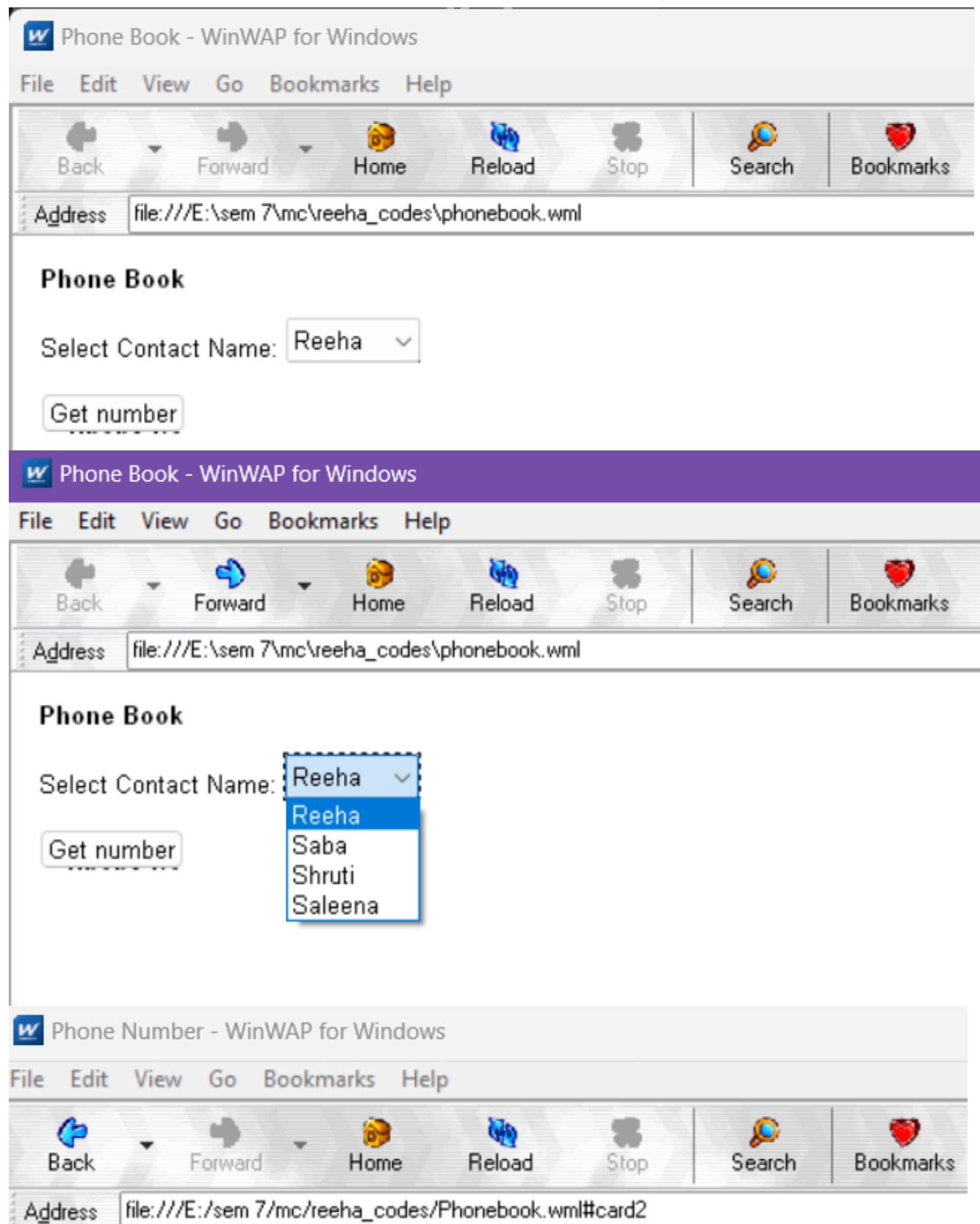
```
extern function phone() {
    var a = WMLBrowser.getVar("name");
    var username = "Full Name : ";
    var email = "Email Id : ";
    var phone = "Contact No : ";

    if(String.compare(a,"Reeha")==0) {
        username = username + "Syeda Reeha Quasar";
        email = email + "Reeha@google.com";
        phone = phone + "9870455572";
    }
    else if(String.compare(a,"Saba")==0) {
        username = username + "Syeda Saba Quasar";
        email = email + "saba@gmail.com";
        phone = phone + "9667678507";
    }
    else if(String.compare(a,"Shruti")==0) {
        username = username + "Shruti Cherwal";
        email = email + "Shruti@gmail.com";
        phone = phone + "9898786765";
    }
    else if(String.compare(a,"Saleena")==0) {
        username = username + "Saleena Fatima";
        email = email + "Saleena@gmail.com";
        phone = phone + "9887867677";
    }
    else {
        username = "Invalid";
        email = "Invalid";
        phone = "Invalid";
    }

    WMLBrowser.setVar("fullname",username);
    WMLBrowser.setVar("phoneNumber",phone);
    WMLBrowser.setVar("email",email);
    WMLBrowser.go("Phonebook.wml#card2");
}
```

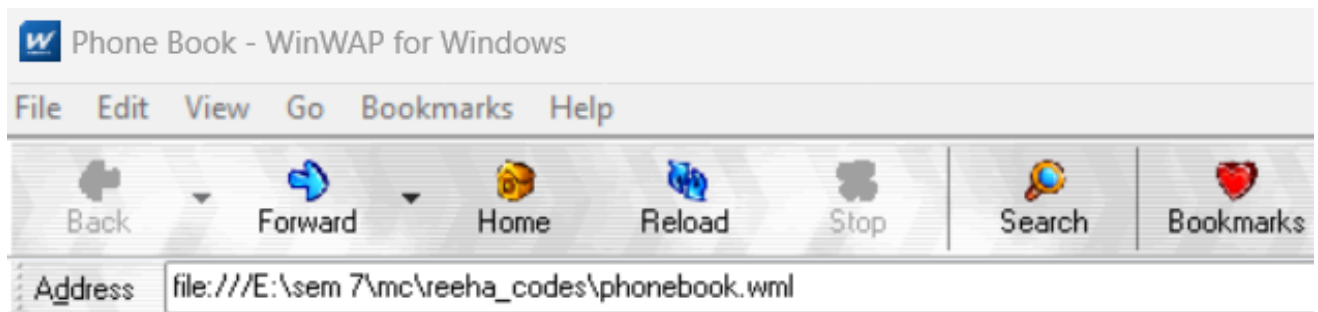


## Output:



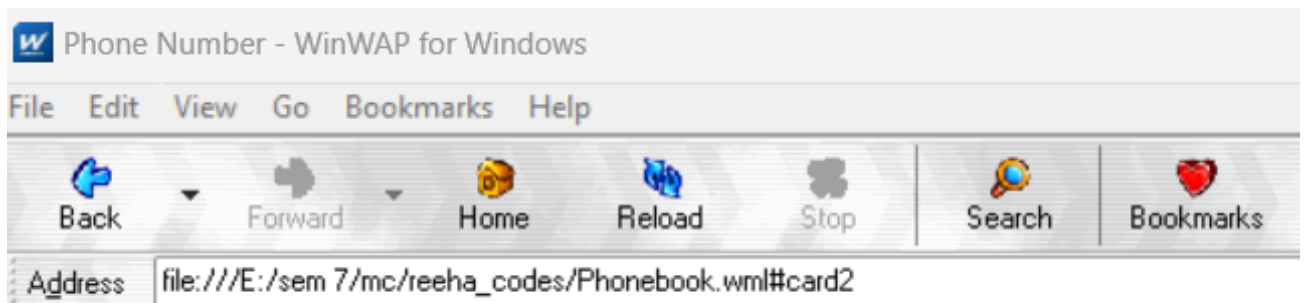
### Contact Details

**Full Name : Syeda Reeha Quasar**  
**Contact No : 9870455572**  
**Email Id : Reeha@google.com**



## Phone Book

Select Contact Name:

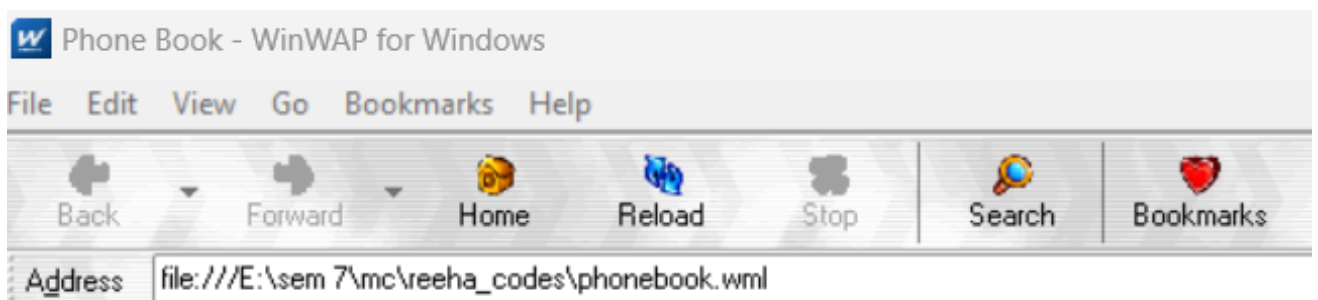


## Contact Details

**Full Name : Syeda Saba Quasar**

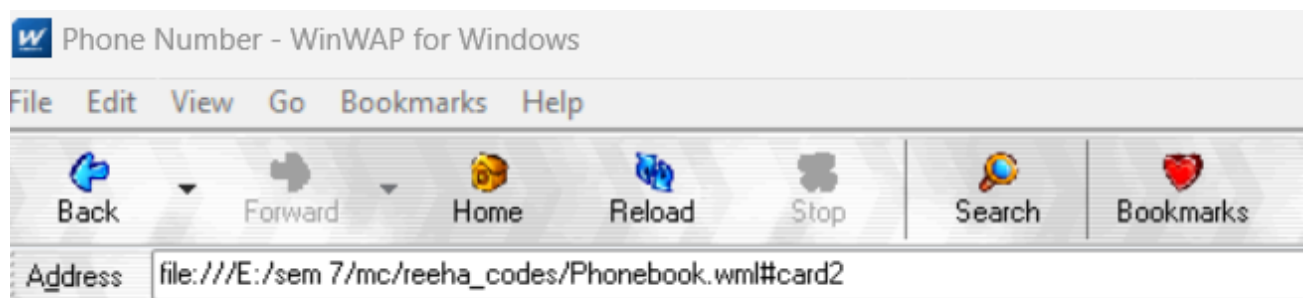
**Contact No : 9667678507**

**Email Id : saba@gmail.com**



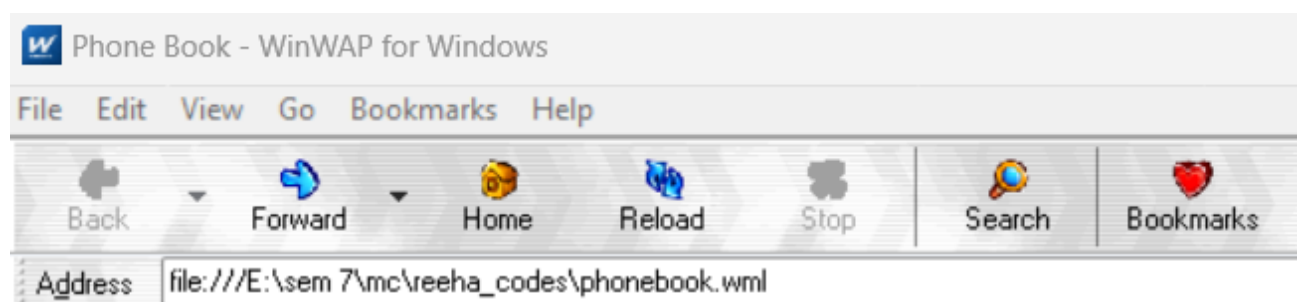
## Phone Book

Select Contact Name:



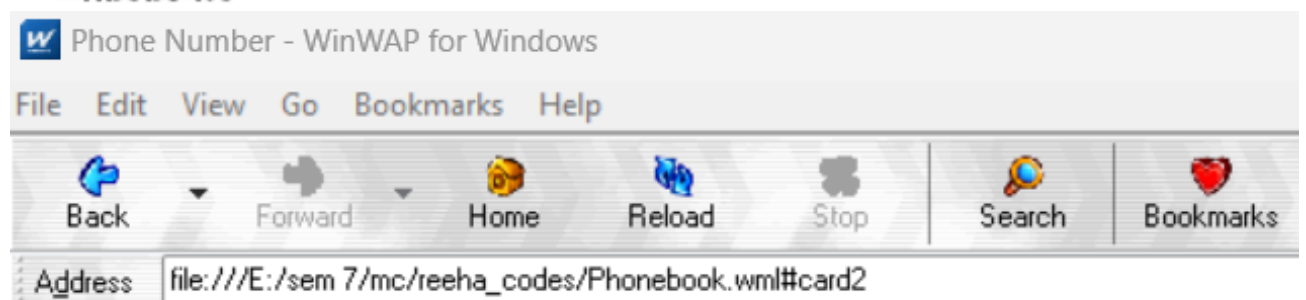
Contact Details

**Full Name : Shruti Cherwal**  
**Contact No : 9898786765**  
**Email Id : Shruti@gmail.com**



### Phone Book

Select Contact Name:



Contact Details

**Full Name : Saleena Fatima**  
**Contact No : 9887867677**  
**Email Id : Saleena@gmail.com**

## **Viva Questions:**

### **Q.1 What is an Activity in Android?**

An Android activity is one screen of the Android app's user interface. In that way an Android activity is very similar to windows in a desktop application. An Android app may contain one or more activities, meaning one or more screens.

### **Q.2 What is an APK format?**

Android Package (APK) is the Android application package file format used by the Android operating system, and a number of other Android-based operating.

### **Q.3 What is an Intent?**

An Intent in the Android operating system is a software mechanism that allows users to coordinate the functions of different activities to achieve a task.

### **Q.4 What is an Android Manifest File?**

The manifest file describes essential information about your app to the Android build tools, the Android operating system, and Google Play. Among many other things, the manifest file is required to declare the following: The app's package name, which usually matches your code's namespace.

# **BEYOND THE SYLLABUS EXPERIMENTS**

## **Experiment 1**

**Aim :** Make an application of your choice using WML or Android.

**Code :**

### **MainActivity.java**

```
package com.shoeARstore;
```

```
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
```

```
import com.daimajia.slider.library.SliderTypes.BaseSliderView;
import com.daimajia.slider.library.SliderTypes.TextSliderView;
```

```
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
```

```
import java.util.ArrayList;
```

```
public class MainActivity extends Fragment implements
BaseSliderView.OnSliderClickListener,NavigationView.OnNavigationItemSelectedListener {
```

```
    public static String id = "id";
    SliderLayout mDemoSlider;
    public static String
server_url="http://konyacaldereliler.com/ShoeArStore/";//http://konyayazibelenkoyu.com/shoea
rstore/ ... http://konyacaldereliler.com/ShoeArStore/ ... http://10.0.2.2/shoearstore/
```

```
SharedPreferences prefs;  
String kullaniciid;
```

```
private ArrayList<Beanlist> Bean;  
private GroomingRecyclerViewAdapter baseAdapter;
```

```
private Context context;  
float positionY;
```

```
Typeface fonts1,fonts2,fonts3,fonts4;  
EditText searchtext;  
ImageView QReader;
```

```
@Nullable  
@Override
```

```
public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container,  
@Nullable Bundle savedInstanceState) {  
    View view=inflater.inflate(R.layout.activity_main,container,false);
```

```
    searchtext = (EditText) view.findViewById(R.id.searchtext);  
    QReader=(ImageView) view.findViewById(R.id.ivQReader);
```

```
    prefs =getActivity().getSharedPreferences("login", Context.MODE_PRIVATE);  
    kullaniciid=prefs.getString("id","0");
```

```
//    *****Listview Header Fontları*****
```

```
    fonts1 = Typeface.createFromAsset(getActivity().getApplicationContext().getAssets(),  
        "fonts/OpenSans-Regular.ttf");  
    fonts2 = Typeface.createFromAsset(getActivity().getApplicationContext().getAssets(),  
        "fonts/OpenSans-Semibold.ttf");
```

```
    fonts3 = Typeface.createFromAsset(getActivity().getApplicationContext().getAssets(),  
        "fonts/Roboto-Medium.ttf");
```

```
    fonts4 = Typeface.createFromAsset(getActivity().getApplicationContext().getAssets(),  
        "fonts/Roboto-Regular.ttf");
```

```
    searchtext.setTypeface(fonts1);
```

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

```
//    *****Slider*****
```

```
    mDemoSlider = (SliderLayout) view.findViewById(R.id.slider);  
    new getData(getActivity()){  
        @Override  
        protected void onPostExecute(String s) {  
            try {
```

```

        if(s!=null){
            JSONArray jsonArray=new JSONArray(s);
            JSONObject jsonObject;
            for (int i=0;i<jsonArray.length();i++)
            {
                TextSliderView textSliderView = new TextSliderView(c);
                jsonObject=jsonArray.getJSONObject(i);
                String path=jsonObject.getString("ImagePath").replace("\\", "/");
                textSliderView
                    // .description(name)
                    .image(server_url+path)
                    .setScaleType(BaseSliderView.ScaleType.CenterInside)
                    .setOnSliderClickListener((BaseSliderView.OnSliderClickListener)
context);

                textSliderView.bundle(new Bundle());
                textSliderView.getBundle().putString("extra",
jsonObject.getString("ImageName"));
                textSliderView.setOnSliderClickListener(new
BaseSliderView.OnSliderClickListener() {
                    @Override
                    public void onSliderClick(BaseSliderView slider) {
                        Toast.makeText(getActivity().getApplicationContext(),
slider.getBundle().getString("extra"), Toast.LENGTH_SHORT).show();
                    }
                });
                mDemoSlider.addSlider(textSliderView);
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    dialog.dismiss();
    super.onPostExecute(s);
}
}.execute(server_url+"slidermaincategory.php","slider","kosulyok");

mDemoSlider.setPresetTransformer(SliderLayout.Transformer.Default);
mDemoSlider.setPresetIndicator(SliderLayout.PresetIndicators.Center_Bottom);
mDemoSlider.setCustomAnimation(new ChildAnimationExample());
mDemoSlider.setDuration(4000);
// *****Kategoriler*****
final LinearLayout layout=(LinearLayout) view.findViewById(R.id.analinearlayout);
final String[] kategori={"En Yeniler","Çok Beğenilenler","Çok Yorumlananlar"};
new getData(getActivity()){
    @Override
    protected void onPostExecute(String s) {
        try {
            if(s!=null){
                JSONArray jsonArray=new JSONArray(s);
                JSONObject jsonObject = null;
                for (int i=0;i<jsonArray.length();i++)
                {
                    jsonObject=jsonArray.getJSONObject(i);
                    final TextView textView=new TextView(getActivity());
                    textView.setText(jsonObject.getString("CategoryName"));

```

```

textView.setId(jsonObject.getInt("Id"));
textView.setPadding(0,25,0,25);
textView.setTextSize(20);
textView.setTypeface(fonts2);
LinearLayout liste=new LinearLayout(getActivity());
liste.setOrientation(LinearLayout.HORIZONTAL);
liste.setPadding(20,0,0,0);
final RecyclerView recyclerView1=new RecyclerView(getActivity());
for(int j=0;j<kategori.length;j++){
    final TextView listeeleman=new TextView(getActivity());
    listeeleman.setText(kategori[j]);
    listeeleman.setId(j);
    if(j==0){
        listeeleman.setBackgroundColor(Color.WHITE);
        final Typeface typeface=Typeface.create("arial-
black",Typeface.BOLD);//sans-serif-light
        listeeleman.setTypeface(typeface);
        listeeleman.setTextColor(Color.parseColor("#24deb6"));
        listeeleman.setY(listeeleman.getY()-12);
    }
    else{
        listeeleman.setBackgroundColor(Color.parseColor("#f2f3f4"));//f2f3f4
        listeeleman.setTextColor(Color.BLACK);
        listeeleman.setTypeface(fonts4);
    }
    listeeleman.setTextSize(15);
    listeeleman.setPadding(10,10,10,10);
    listeeleman.setOnClickListener(new View.OnClickListener() {
        @SuppressWarnings("ResourceType")
        @Override
        public void onClick(View v) {
            kontrol(v.getId(),v);
            adapterGonder(textView.getId(),((TextView)
v).getText().toString(),recyclerView1);
        }
    });
    liste.addView(listeeleman);
    liste.invalidate();
    positionY=listeeleman.getY();
}
adapterGonder(jsonObject.getInt("Id"),kategori[0],recyclerView1);
recyclerView1.setHasFixedSize(true);
recyclerView1.setId(i+1);
LinearLayoutManager rManager=new
LinearLayoutManager(getActivity(),LinearLayoutManager.HORIZONTAL,false);
recyclerView1.setLayoutManager(rManager);
layout.addView(textView);
layout.addView(liste);
layout.addView(recyclerView1);
layout.invalidate();
}
}
} catch (JSONException e) {
    e.printStackTrace();
}
}

```



```

        dialog.dismiss();
        super.onPostExecute(s);
    }
    }.execute(server_url+"category.php","maincategory","kosulyok");
    return view;
}

private void adapterGonder(int id, String subcategory, final RecyclerView recyclerView) {
    String kosul = null;
    if(subcategory.equals("En Yeniler")){//Çok Satanlar
        kosul="DateOrder";
    }else if(subcategory.equals("En Çok Beğenilenler")){//En çok beğenilenler
        kosul="RatingOrder";
    }else if(subcategory.equals("En Çok Yorumlananlar")){//En Çok Yorumlananlar
        kosul="CommentOrder";
    }
    new getData(getActivity()){
        @Override
        protected void onPostExecute(String s) {
            if(s!=null) {
                ArrayList<String> image=new ArrayList<>();
                ArrayList<String> name=new ArrayList<>();
                ArrayList<String> price=new ArrayList<>();
                ArrayList<String> cutprice=new ArrayList<>();
                ArrayList<String> discount=new ArrayList<>();
                ArrayList<String> ratingtext=new ArrayList<>();
                ArrayList<String> rating=new ArrayList<>();
                ArrayList<String> id=new ArrayList<>();
                ArrayList<String> favorites=new ArrayList<>();
                JSONArray jsonArray = null;
                try {
                    JSONObject jsonObject=null;
                    jsonArray=new JSONArray(s);
                    for(int i=0;i<javascriptArray.length();i++){
                        jsonObject=jsonArray.getJSONObject(i);
                        String path=jsonObject.getString("Resim").replace("\\", "/");
                        image.add(server_url+path);
                        name.add(jsonObject.getString("ProductName"));
                        price.add(jsonObject.getString("LastPrice"));
                        cutprice.add(jsonObject.getString("FirstPrice"));
                        float fark=Float.parseFloat(jsonObject.getString("FirstPrice"))-
Float.parseFloat(jsonObject.getString("LastPrice"));
                        int indirim= (int)
((fark*100)/Float.parseFloat(jsonObject.getString("FirstPrice")));
                        discount.add("%"+String.valueOf(indirim));
                        ratingtext.add(jsonObject.getString("Comment"));
                        if(jsonObject.getString("Rating").equals("null")){
                            rating.add("0");
                        }else{
                            rating.add(jsonObject.getString("Rating"));
                        }
                        favorites.add(jsonObject.getString("Favorites"));
                        id.add(jsonObject.getString("StockCode"));
                    }
                }
            }
        }
    }
}

```

```

gonder(image,name,price,cutprice,discount,ratingtext,rating,id,favorites,recyclerView);
        dialog.dismiss();
    } catch (JSONException e) {
        e.printStackTrace();
    }
} else {
    baseAdapter=null;
}
dialog.dismiss();
super.onPostExecute(s);
}
}.execute(server_url+"sirala.php",String.valueOf(id),kullaniciid+"-"+kosul);
}

```

```

private void gonder(ArrayList<String> image, ArrayList<String> title, ArrayList<String>
price, ArrayList<String> cutprice, ArrayList<String> discount
    , ArrayList<String> ratingtext, ArrayList<String> rating, ArrayList<String> id,
ArrayList<String> favorites, RecyclerView recyclerView) {
    Bean=new ArrayList<Beanlist>();
    for(int i=0;i<image.size();i++){
        Beanlist Beanlist = new Beanlist(image.get(i)
            , title.get(i), price.get(i), cutprice.get(i), discount.get(i), ratingtext.get(i)
            ,Float.parseFloat(rating.get(i)) ,id.get(i),favorites.get(i));
        Bean.add(Beannlist);
    }
    baseAdapter = new GroomingRecyclerViewAdapter(getActivity(), getActivity(), Bean) {
};

```

```

    if(baseAdapter!=null) {
        baseAdapter.setOnItemClickListener(new
GroomingRecyclerViewAdapter.MyClickListener() {
            @Override
            public void onItemClick(int position, View v) {
                TextView id = (TextView) v.findViewById(R.id.id);
                Intent gec = new Intent(getActivity(), ProductDetailActivity.class);
                gec.putExtra("StockCode",id.getText().toString());
                startActivity(gec);
            }
        });
    }
}

```

```

recyclerView.setAdapter(baseAdapter);
}

```

```

private void kontrol(int id, View v) {
    LinearLayout parent= (LinearLayout) v.getParent();
    for (int i=0;i<parent.getChildCount();i++){
        TextView textView=(TextView) parent.getChildAt(i);
        if(textView.getId()==id){
            final Typeface typeface=Typeface.create("arial-
black",Typeface.BOLD_ITALIC);//sans-serif-light
            textView.setBackgroundColor(Color.WHITE);
            textView.setTypeface(typeface);
            textView.setTextColor(Color.parseColor("#24deb6"));
            textView.setY(positionY-12);

```

```

        }else{
            textView.setTextColor(Color.BLACK);
            textView.setBackgroundColor(Color.parseColor("#f2f3f4"));
            textView.setY(positionY);
            textView.setTypeface(fonts4);
        }
    }
}

public static MainActivity newInstance(){
    MainActivity fragment=new MainActivity();
    return fragment;
}

@Override
public void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

@Override
public void onSliderClick(BaseSliderView slider) {
}

@Override
public boolean onNavigationItemSelected(MenuItem item) {
    return false;
}
}

```

### **Main\_activity.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#f4f4f4"
    tools:context=".ProductListActivity">

    <include
        layout="@layout/toolbarproductlist"
        android:id="@+id/toolbar"
    ></include>

    <LinearLayout
        android:layout_below="@+id/toolbar"
        android:id="@+id/linear1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/toolbarbg"
        android:paddingBottom="20dp"
        android:paddingTop="10dp"
        android:orientation="horizontal"

```

>

```
<Spinner
    android:id="@+id/sirala"
    android:layout_weight="1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="SIRALA"
    android:drawableLeft="@drawable/arrowupdown"
    android:layout_gravity="center"
    android:gravity="center"
    android:layout_marginLeft="5dp"
    android:textColor="@color/Text"/>
```

<View

```
    android:layout_width="1dp"
    android:layout_height="30dp"
    android:background="@color/colordivider"/>
<customfonts.MyTextView
    android:layout_weight="1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="FILTRELE"
    android:drawableLeft="@drawable/refine"
    android:drawablePadding="10dp"
    android:layout_gravity="center"
    android:gravity="center"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="15dp"
    android:textColor="@color/Text"
/>
```

<View

```
    android:layout_width="1dp"
    android:layout_height="30dp"
    android:layout_marginRight="30dp"
    android:background="@color/colordivider"/>
```

<ImageView

```
    android:id="@+id/gridviewicon"
    android:visibility="visible"
    android:layout_marginRight="20dp"
    android:layout_width="15dp"
    android:layout_height="15dp"
    android:layout_gravity="center"
    android:background="@drawable/gridviewicon"/>
```

<ImageView

```
    android:id="@+id/listviewicon"
    android:visibility="gone"
    android:layout_marginRight="20dp"
    android:layout_width="15dp"
    android:layout_height="15dp"
    android:layout_gravity="center"
```

```
android:background="@drawable/listviewicon"/>
```

```
</LinearLayout>
```

```
<ScrollView
```

```
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_marginTop="103dp"  
    android:scrollbars="none"  
>
```

```
<RelativeLayout
```

```
    android:layout_marginTop="15dp"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">
```

```
<com.shoeARstore.ExpandableHeightGridView
```

```
    android:visibility="visible"  
    android:layout_height="wrap_content"  
    android:layout_width="wrap_content"  
    android:verticalSpacing="5dp"  
    android:stretchMode="columnWidth"  
    android:numColumns="2"  
    android:layout_margin="5dp"  
    android:isScrollContainer="false"  
    android:horizontalSpacing="5dp"  
    android:id="@+id/gridview"  
    android:background="#f2f3f4" />
```

```
<com.shoeARstore.ExpandableHeightListView
```

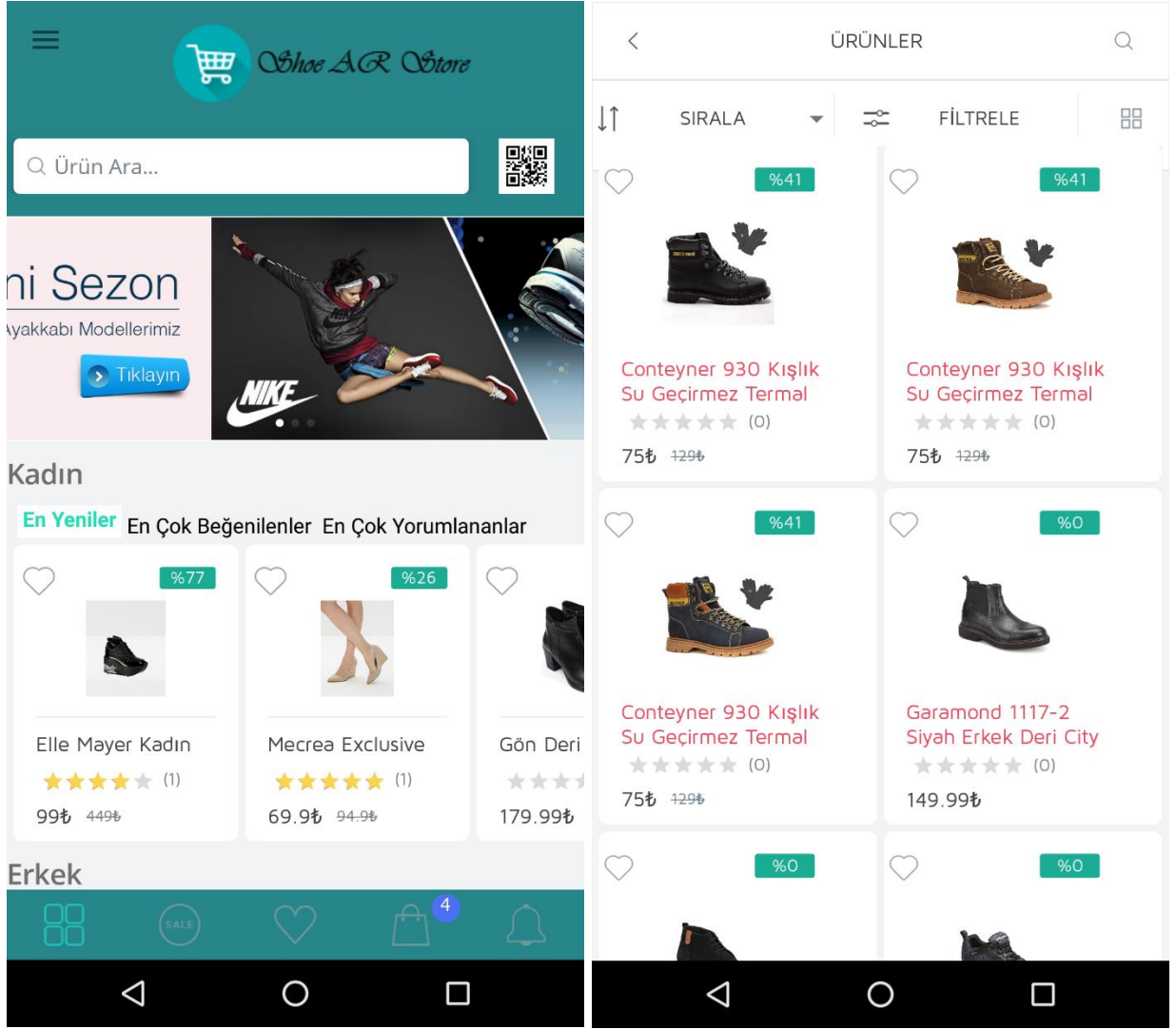
```
    android:layout_marginTop="15dp"  
    android:visibility="gone"  
    android:id="@+id/listview"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:background="#f2f3f4"/>
```

```
</RelativeLayout>
```

```
</ScrollView>
```

```
</RelativeLayout>
```

## Output:



## **Viva Questions:**

### **Q.1. What is an Activity in Android?**

Android activity is a single, focused thing that the user can do. Activities are often presented to the user as a single screen on a mobile device with a task-oriented user interface. Activities can be launched by the user or by other components of your application.

### **Q.2. What is an APK format?**

APK file is an Android application package file format used for distribution and installation of mobile apps on the Android operating system. APK files are based on the ZIP file format and contain all of the code, resources, and other data needed to run an app on an Android device.

### **Q.3. What is an Intent?**

Intent is an object that represents an action to be performed. Intents are used to start activities, services, broadcast receivers, and content providers. They can also be used to pass data between components of an application.

### **Q.4. What is an Android Manifest File?**

The Android manifest file is a file that describes the components of your Android application. The manifest file is used by the Android system to find and load your application's components. The manifest file also contains information about your application's permissions, features, and other settings.

## Experiment 2

**Aim :** Program to develop a calling application.

**Code :**

### MainActivity.java

```
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    // define objects for edit text and button
    EditText edittext;
    Button button;

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

        // Getting instance of edittext and button
        button = findViewById(R.id.button);
        edittext = findViewById(R.id.editText);

        // Attach set on click listener to the button for initiating intent
        button.setOnClickListener(arg -> {
            // getting phone number from edit text and changing it to String
            String phone_number = edittext.getText().toString();

            // Getting instance of Intent with action as ACTION_CALL
            Intent phone_intent = new Intent(Intent.ACTION_CALL);

            // Set data of Intent through Uri by parsing phone number
            phone_intent.setData(Uri.parse("tel:" + phone_number));

            // start Intent
            startActivity(phone_intent);
        });
    }
}
```



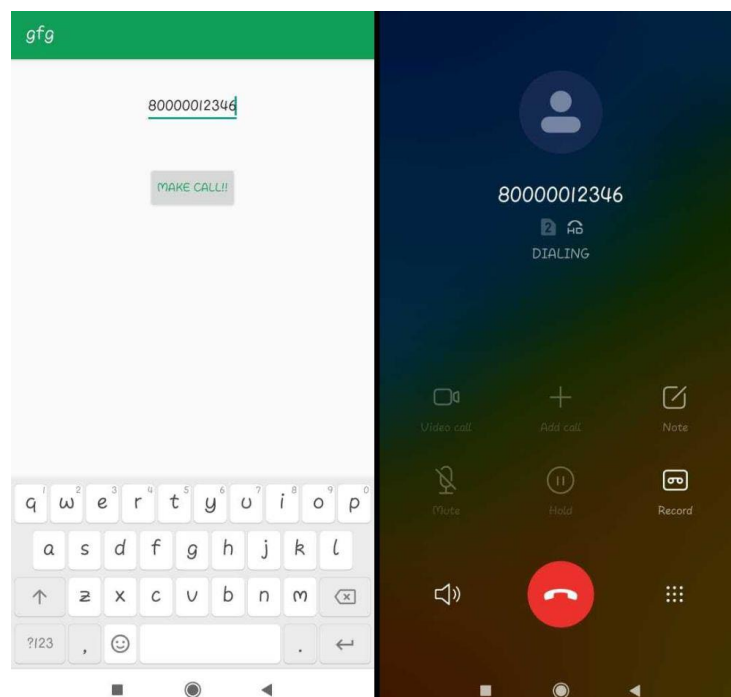
## activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!--Relative Layout-->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!-- Edit text for phone number -->
    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="30dp" />

    <!-- Button to make call -->
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="115dp"
        android:padding="5dp"
        android:text="Make Call!!" />
</RelativeLayout>
```

## Output:





## Create Meet



## **Viva Questions:**

### **Q.1. What is the Android SDK?**

The Android SDK (Software Development Kit) is a set of tools and libraries that you need to develop Android applications. The SDK includes a compiler, debugger, emulator, and a variety of other tools that help you write, test, and debug your applications.

### **Q.2. What is the Android Studio IDE?**

Android Studio is an integrated development environment (IDE) for developing Android applications. Android Studio is based on IntelliJ IDEA, and it provides a number of features that make it easier to develop Android applications, such as code completion, refactoring, and debugging tools.

### **Q.3. What are the different types of Android layouts?**

There are three main types of Android layouts:

- Linear layouts: Linear layouts arrange widgets in a single row or column.
- Relative layouts: Relative layouts arrange widgets relative to each other.
- Frame layouts: Frame layouts contain a single widget, and they can be used to create a floating window or a dialog box.

### **Q.4. What are the different types of Android widgets?**

There are many different types of Android widgets, but some of the most common ones include:

- Buttons: Buttons are used to perform actions, such as launching an activity or opening a dialog box.
- TextViews: TextViews are used to display text.
- EditTexts: EditTexts are used to allow users to enter text.
- ImageViews: ImageViews are used to display images.

### **Q.5. What are the different types of Android intents?**

There are many different types of Android intents, but some of the most common ones include:

- Explicit intents: Explicit intents are used to start a specific activity.
- Implicit intents: Implicit intents are used to start an activity that can handle a specific action.
- Broadcast intents: Broadcast intents are used to send a message to all interested components.

## Experiment 3

**Aim :** Program to develop a mailing application

### Source Code :

#### MainActivity.java

```
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    // define objects for edit text and button
    Button button;
    EditText sendto, subject, body;

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

        // Getting instance of edittext and button
        sendto = findViewById(R.id.editText1);
        subject = findViewById(R.id.editText2);
        body = findViewById(R.id.editText3);
        button = findViewById(R.id.button);

        // attach setOnClickListener to button with Intent object define in it
        button.setOnClickListener(view -> {
            String emailsend = sendto.getText().toString();
            String emailsubject = subject.getText().toString();
            String emailbody = body.getText().toString();

            // define Intent object with action attribute as ACTION_SEND
            Intent intent = new Intent(Intent.ACTION_SEND);

            // add three fields to intent using putExtra function
            intent.putExtra(Intent.EXTRA_EMAIL, new String[]{emailsend});
            intent.putExtra(Intent.EXTRA_SUBJECT, emailsubject);
            intent.putExtra(Intent.EXTRA_TEXT, emailbody);

            // set type of intent
            intent.setType("message/rfc822");

            // startActivity with intent with chooser as Email client using createChooser
            startActivity(Intent.createChooser(intent, "Choose an Email client :"));
        });
    }
}
```

```
}
```

## activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Relative Layout -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!-- Edit text for email id -->
    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_marginTop="18dp"
        android:layout_marginRight="22dp" />

    <!-- Edit text for email subject -->
    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText1"
        android:layout_alignLeft="@+id/editText1"
        android:layout_marginTop="20dp" />

    <!-- Edit text for email body -->
    <EditText
        android:id="@+id/editText3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText2"
        android:layout_alignLeft="@+id/editText2"
        android:layout_marginTop="30dp" />

    <!-- text Views for label -->
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText1"
        android:layout_alignBottom="@+id/editText1"
        android:layout_alignParentLeft="true"
        android:text="Send To:"
        android:textColor="#0F9D58" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

        android:layout_alignBaseline="@+id/editText2"
        android:layout_alignBottom="@+id/editText2"
        android:layout_alignParentLeft="true"
        android:text="Email Subject:"
        android:textColor="#0F9D58" />

```

<TextView

```

        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText3"
        android:layout_alignBottom="@+id/editText3"
        android:text="Email Body:"
        android:textColor="#0F9D58" />

```

<!-- Button to send email -->

<Button

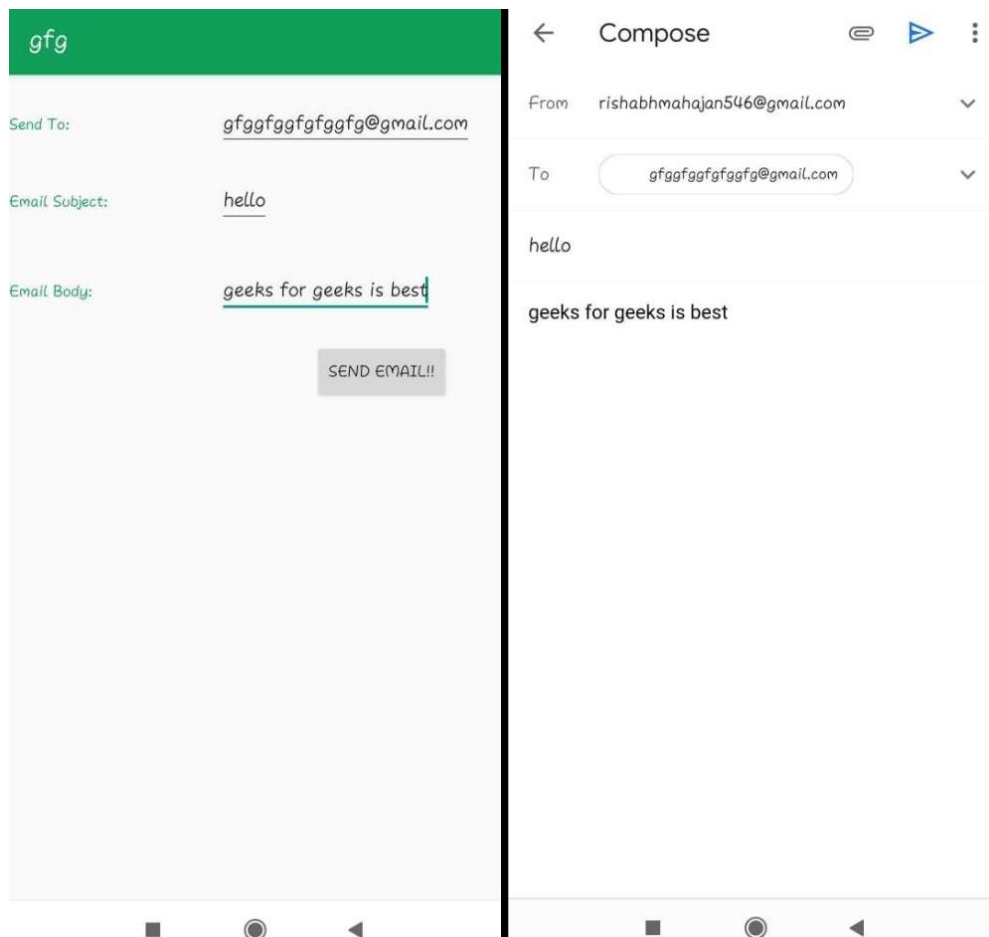
```

        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText3"
        android:layout_alignLeft="@+id/editText3"
        android:layout_marginLeft="76dp"
        android:layout_marginTop="20dp"
        android:text="Send email!!" />

```

</RelativeLayout>

## Output:



## SendEmailApp

Recipient

subject

Message Body

SEND MAIL

## **Viva Questions:**

### **Q1: What are the different types of calls that can be made in Android?**

There are two main types of calls that can be made in Android:

- Outgoing calls: These are calls that are initiated by the user.
- Incoming calls: These are calls that are received by the user.

In addition to these two types of calls, there are also a number of other types of calls that can be made in Android, such as:

- Conference calls: These are calls that involve multiple participants.
- Video calls: These are calls that involve both audio and video.
- Emergency calls: These are calls that are made to emergency services, such as 911.

### **Q2: How do I make an outgoing call in Android?**

To make an outgoing call in Android, you can use the following steps:

- 1) Open the Phone app.
- 2) Enter the phone number that you want to call.
- 3) Tap the Call button.

If you have the phone number saved in your contacts, you can also tap on the contact's name to call them.

### **Q3: How do I answer an incoming call in Android?**

To answer an incoming call in Android, you can use the following steps:

- 1) When you receive an incoming call, a notification will appear on your screen.
- 2) Tap on the Answer button to answer the call.

You can also answer an incoming call by pressing the Volume Up button on your device.

### **Q4: How do I reject an incoming call in Android?**

Answer: To reject an incoming call in Android, you can use the following steps:

- 1) When you receive an incoming call, a notification will appear on your screen.
- 2) Tap on the Decline button to reject the call.

You can also reject an incoming call by pressing the Power button on your device.

### **Q5: How do I mute an incoming call in Android?**

To mute an incoming call in Android, you can use the following steps:

- 1) When you receive an incoming call, a notification will appear on your screen.
- 2) Tap on the Mute button to mute the call.

You can also mute an incoming call by pressing the Volume Down button on your device.



## Experiment 4

**Aim :** Program to design a contact application.

### Source Code :

#### MainActivity.java

```
import android.Manifest;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.provider.Settings;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.ProgressBar;
import android.widget.SearchView;
import android.widget.Toast;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.MenuItemCompat;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.karumi.dexter.Dexter;
import com.karumi.dexter.MultiplePermissionsReport;
import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.DexterError;
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.PermissionRequestErrorListener;
import com.karumi.dexter.listener.multi.MultiplePermissionsListener;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

    // creating variables for our array list, recycler view progress bar and adapter.
    private ArrayList<ContactsModal> contactsModalArrayList;
    private RecyclerView contactRV;
    private ContactRVAdapter contactRVAdapter;
    private ProgressBar loadingPB;

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

```

setContentView(R.layout.activity_main);

// on below line we are initializing our variables.
contactsModalArrayList = new ArrayList<>();
contactRV = findViewById(R.id.idRVContacts);
FloatingActionButton addNewContactFAB = findViewById(R.id.idFABAdd);
loadingPB = findViewById(R.id.idPBLoading);

// calling method to prepare our recycler view.
prepareContactRV();

// calling a method to request permissions.
requestPermissions();

// adding on click listener for our fab.
addNewContactFAB.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // opening a new activity on below line.
        Intent i = new Intent(MainActivity.this,
CreateNewContactActivity.class);
        startActivity(i);
    }
});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // in this on create options menu we are calling
    // a menu inflater and inflating our menu file.
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.search_menu, menu);
    // on below line we are getting our menu item as search view item
    MenuItem searchViewItem = menu.findItem(R.id.app_bar_search);
    // on below line we are creating a variable for our search view.
    final SearchView searchView = (SearchView)
MenuItemCompat.getActionView(searchViewItem);
    // on below line we are setting on query text listener for our search view.
    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String query) {
            // on query submit we are clearing the focus for our search view.
            searchView.clearFocus();
            return false;
        }

        @Override
        public boolean onQueryTextChange(String newText) {
            // on changing the text in our search view we are calling
            // a filter method to filter our array list.
            filter(newText.toLowerCase());
            return false;
        }
    });
    return super.onCreateOptionsMenu(menu);
}

```

```

    }

    private void filter(String text) {
        // in this method we are filtering our array list.
        // on below line we are creating a new filtered array list.
        ArrayList<ContactsModal> filteredlist = new ArrayList<>();
        // on below line we are running a loop for checking if the item is present in array list.
        for (ContactsModal item : contactsModalArrayList) {
            if (item.getUserName().toLowerCase().contains(text.toLowerCase())) {
                // on below line we are adding item to our filtered array list.
                filteredlist.add(item);
            }
        }
        // on below line we are checking if the filtered list is empty or not.
        if (filteredlist.isEmpty()) {
            Toast.makeText(this, "No Contact Found",
Toast.LENGTH_SHORT).show();
        } else {
            // passing this filtered list to our adapter with filter list method.
            contactRVAdapter.filterList(filteredlist);
        }
    }

    private void prepareContactRV() {
        // in this method we are preparing our recycler view with adapter.
        contactRVAdapter = new ContactRVAdapter(this, contactsModalArrayList);
        // on below line we are setting layout manager.
        contactRV.setLayoutManager(new LinearLayoutManager(this));
        // on below line we are setting adapter to our recycler view.
        contactRV.setAdapter(contactRVAdapter);
    }

    private void requestPermissions() {
        // below line is use to request
        // permission in the current activity.
        Dexter.withActivity(this)
            // below line is use to request the number of
            // permissions which are required in our app.
            .withPermissions(Manifest.permission.READ_CONTACTS,
                // below is the list of permissions
                Manifest.permission.CALL_PHONE,
                Manifest.permission.SEND_SMS,
Manifest.permission.WRITE_CONTACTS)
            // after adding permissions we are
            // calling and with listener method.
            .withListener(new MultiplePermissionsListener() {
                @Override
                public void
onPermissionsChecked(MultiplePermissionsReport multiplePermissionsReport) {
                    // this method is called when all permissions are
                    granted
                    if
(multiplePermissionsReport.areAllPermissionsGranted()) {
                        // do you work now
                        getContacts();
                    }
                }
            })
    }

```

```

        Toast.makeText(MainActivity.this, "All the
permissions are granted..", Toast.LENGTH_SHORT).show();
    }
    // check for permanent denial of any permission
    if
(multiplePermissionsReport.isAnyPermissionPermanentlyDenied()) {
        // permission is denied permanently,
        // we will show user a dialog message.
        showSettingsDialog();
    }
}

@Override
public void
onPermissionRationaleShouldBeShown(List<PermissionRequest> list, PermissionToken
permissionToken) {

    // this method is called when user grants some
    // permission and denies some of them.
    permissionToken.continuePermissionRequest();
}

}).withErrorListener(new PermissionRequestErrorListener() {
// this method is use to handle error
// in runtime permissions
@Override
public void onError(DexterError error) {
    // we are displaying a toast message for error message.
    Toast.makeText(getApplicationContext(), "Error occurred! ",
Toast.LENGTH_SHORT).show();
}
})

// below line is use to run the permissions
// on same thread and to check the permissions
.onSameThread().check();
}

// below is the shoe setting dialog
// method which is use to display a
// dialogue message.
private void showSettingsDialog() {
    // we are displaying an alert dialog for permissions
    AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

    // below line is the title
    // for our alert dialog.
    builder.setTitle("Need Permissions");

    // below line is our message for our dialog
    builder.setMessage("This app needs permission to use this feature. You can grant
them in app settings.");
    builder.setPositiveButton("GOTO SETTINGS", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            // this method is called on click on positive
            // button and on clicking shit button we

```

```

        // are redirecting our user from our app to the
        // settings page of our app.
        dialog.cancel();
        // below is the intent from which we
        // are redirecting our user.
        Intent intent = new
Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS);
        Uri uri = Uri.fromParts("package", getPackageName(), null);
        intent.setData(uri);
        startActivityForResult(intent, 101);
    }
});
builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        // this method is called when
        // user click on negative button.
        dialog.cancel();
    }
});
// below line is used
// to display our dialog
builder.show();
}

private void getContacts() {
    // this method is use to read contact from users device.
    // on below line we are creating a string variables for
    // our contact id and display name.
    String contactId = "";
    String displayName = "";
    // on below line we are calling our content resolver for getting contacts
    Cursor cursor =
getContentResolver().query(ContactsContract.Contacts.CONTENT_URI, null, null, null,
ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME + " ASC");
    // on blow line we are checking the count for our cursor.
    if (cursor.getCount() > 0) {
        // if the count is greater than 0 then we are running a loop to move our cursor
to next.
        while (cursor.moveToNext()) {
            // on below line we are getting the phone number.
            int hasPhoneNumber =
Integer.parseInt(cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts.HAS_PH
ONE_NUMBER)));
            if (hasPhoneNumber > 0) {
                // we are checking if the has phone number is > 0
                // on below line we are getting our contact id and user name
for that contact
                contactId =
cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts._ID));
                displayName =
cursor.getString(cursor.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));
                // on below line we are calling a content resolver and making a
query
                Cursor phoneCursor = getContentResolver().query(

```

```

ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
                                null,

ContactsContract.CommonDataKinds.Phone.CONTACT_ID + " = ?",
                                new String[]{contactId},
                                null);
        // on below line we are moving our cursor to next position.
        if (phoneCursor.moveToNext()) {
            // on below line we are getting the phone number for
our users and then adding the name along with phone number in array list.
            String phoneNumber =
phoneCursor.getString(phoneCursor.getColumnIndex(ContactsContract.CommonDataKinds.Ph
one.NUMBER));

                                contactsModalArrayList.add(new
ContactsModal(displayName, phoneNumber));
        }
        // on below line we are closing our phone cursor.
        phoneCursor.close();
    }
}
// on below line we are closing our cursor.
cursor.close();
// on below line we are hiding our progress bar and notifying our adapter class.
loadingPB.setVisibility(View.GONE);
contactRVAdapter.notifyDataSetChanged();
}
}

```

### **activity\_main.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <!--Recycler view for displaying list of contacts-->
    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/idRVContacts"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <!--progress bar for displaying loading-->
    <ProgressBar
        android:id="@+id/idPBLoading"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true" />

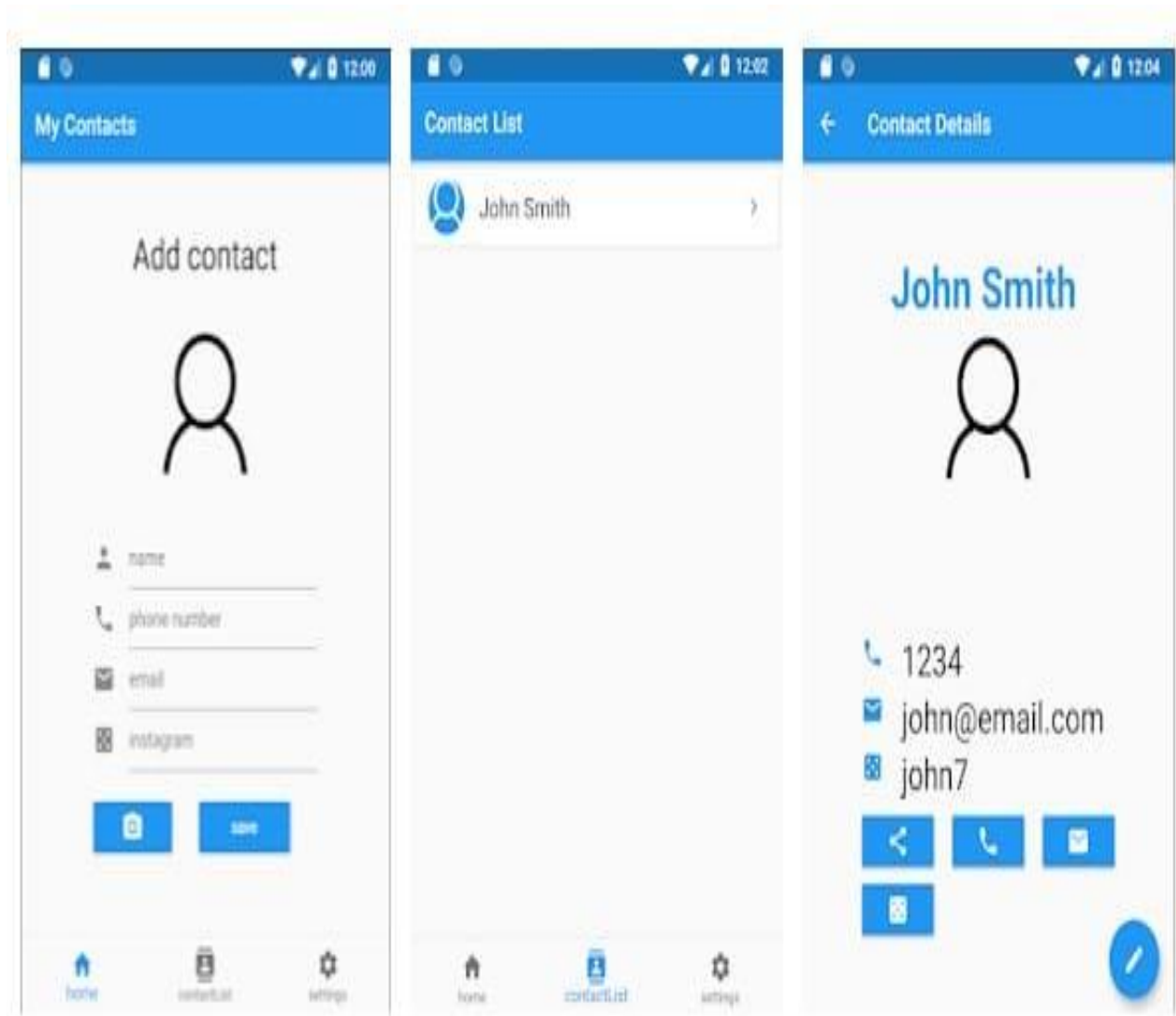
```

<!--fab for adding a new contact-->

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/idFABadd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_margin="20dp"
    android:src="@drawable/ic_account"
    app:fabCustomSize="40dp"
    app:tint="@color/white" />
```

</RelativeLayout>

## Output:



## Viva Questions:

### Q.1. What is an XML file?

An XML file is a text file that uses a special markup language to define the structure of data. XML is a popular format for storing data, and it is used by a wide variety of applications, including Android applications.

### Q.2. What are the benefits of using XML in Android applications?

There are a number of benefits to using XML in Android applications, including:

- Portability: XML is a standard format, so it can be used to store data that can be easily transferred between different applications and platforms.
- Flexibility: XML is a flexible format, so it can be used to store a wide variety of data types.
- Efficiency: XML files are typically smaller than other types of data files, such as binary files. This can make them faster to load and save.

### Q.3. What are the different types of XML elements?

There are two main types of XML elements:

- Start tags: Start tags define the beginning of an XML element. They have the following format:
  - `<element_name>`
- End tags: End tags define the end of an XML element. They have the following format:
  - `</element_name>`

### Q.4. What are attributes?

Attributes are additional pieces of information that can be associated with an XML element. They are defined within the start tag of the element, and they have the following format:

**`attribute_name="attribute_value"`**

### Q.5. How do I create an XML file in Android Studio?

To create an XML file in Android Studio, you can use the following steps:

- 1) In the Project window, right-click on the folder where you want to create the file and select "New > File".
- 2) In the "New File" dialog, select the "XML" file type and click "Next".
- 3) In the "File Name" field, enter the name of your file.
- 4) In the "Location" field, select the folder where you want to save the file.
- 5) Click "Finish".

Your new XML file will be created in the selected folder.