

TERNA ENGINEERING COLLEGE
Computer Engineering

Lab Manual

Of

Mobile Computing- (CSL 603)

T.E. COMP (A,B&C)

SEM VI

F. H. 2025

Lab Objectives

- 1) To understand basics of cellular system.
- 2) **To introduce students with various tools like** Android Studio, NS2, Wire-shark, Cisco packet tracer, WAP supported browser etc.
- 3) To understand the security algorithms in mobile networks.

Lab Outcomes

- 1) Students will able to develop and demonstrate mobile applications using various tools.
- 2) Student will be able to articulate the knowledge of GSM, CDMA & Bluetooth technologies and demonstrate it.
- 3) Students will able to carry out simulation of frequency reuse, hidden/exposed terminal.
- 4) Student will be able to implement security algorithms for mobile communication network.
- 5) Student will be able demonstrate simulation and compare the performance of Wireless LAN.
- 6) Student will be able to develop mobile app using flutter/Android Studio

PSO's

- 1) PSO1:** Inculcate skills to recognize, analyze the problems related to databases, computing, networks and any other domain specific application and provide solutions.
- 2) PSO2:** Ability to develop efficient, secure, user friendly and cost effective software systems.

Experiment List

Exp. No.	Experiment	Type	Language/ Tools used.	LO Mapping	PO Mapping	PSO 1	PSO2
1	To understand the cellular frequency reuse concept to find the co-channel cells for a particular cell	Design and analysis and implementation	Virtual lab, iit kharagpur	3	1,2,3,4, 5,12	2	1
2	To implement a basic function of Code Division Multiple Access (CDMA) to test the orthogonality and autocorrelation of a code to be used for CDMA operation. Write an application based on the above concept.	Design and analysis and implementation	Java, Python	2	1,2,4,5, 10,12	2	2
3	Implementation of GSM security algorithms (A3/A5/A8)	Design and analysis and implementation	Java, Python	4	1,2,3,4, 5,10,1 2	2	2
4	Configure and set up access point. Design WLAN.	Design and analysis and implementation	Cisco packet tracer	5	1,2,3,4, 5,10,1 2	3	2
5	Implementation a Bluetooth network with application as transfer of a File from one device to another.	Design and analysis and implementation	Python	2	1,2,3,4 ,5,10,1 2	2	2
6	To develop an android application that creates an alert upon receiving a message	Design and analysis and implementation	Android studio	1,6	1,2,4,5 ,10,12	2	2
7	To develop an android application that uses GPS Location Information	Design and analysis and implementation	Android studio	1,6	1,2,4,5 ,10,12	2	2
8	To Develop an application that uses GUI components, Fonts and colors	Design and analysis and implementation	Android studio	1,6	1,2,4,5 ,10,12	2	2

9	To Write an application that draws basic graphical primitives in the screen	Design and analysis and implementation	Android studio	1,6	1 ,2,4, 5 ,10,12	2	2
10	To Implementation of income tax/loan EMI calculator and deploy the same on real devices	Design and analysis and implementation	Android studio	1,6	1 ,2,4, 5 ,10,12	2	2

Experiment No-1:-Frequency reuse

PART A

(PART A: TO BE REFERRED BY STUDENTS)

A.1 Aim

To understand the cellular network frequency reuse concept fulfilling the following objectives:

1. Finding the co-channel cells for a particular cell.
2. Finding the cell clusters within certain geographic area.

A.2. Objectives: To understand basics of cellular system

A.3. Outcomes: Students will able to carry out simulation of frequency reuse, hidden/exposed terminal.(LO-3)

A.4 Theory:

- In mobile communication systems a slot of a carrier frequency / code in a carrier frequency is a **radio resource unit**.
- This radio resource unit is assigned to a user in order to support a call/ session. **The number of available such radio resources at a base station thus determines the number of users who can be supported in the call.**
- Since in wireless channels a signal is "broadcast" i.e. received by all entities therefore one a resource is allocated to a user's it cannot be re-assigned until the user finished the call/ session. **Thus the number of users who can be supported in a wireless system is highly limited.**
- **In order to support a large no. of users within a limited spectrum in a region the concept of frequency re-use is used.**
- **The signal radiated from the transmitter antenna gets attenuated** with increasing distance. At a certain distance the signal strength falls below noise threshold and **is no longer identifiable.**

- In this region when the signal attenuates below noise floor the same radio resource may be used by another transmission to send different information.
- In term of cellular systems, the same radio resource (frequency) **can use by two base stations which are sufficiently spaced apart**. In this way the same frequency gets reused in a layer- geographic area **by two or more different base stations** different users simultaneously.
- Now what is important is to select the set of **base stations** which will use the same set of radio resources/ channel of frequencies or **technically the co-channel cells**.
- In this context the minimum adjacent set cells which use different frequencies each is called a cluster.
- The cellular concept is the major solution of the problem of spectral congestion and user capacity. Cellular radiorely on an intelligent allocation and channel reuse throughout a large geographical coverage region.

+ **Cellular Frequency Reuse:**

- Each cellular base station is allocated a **group of radio channels** to be used within a small geographic area called a cell.
- Base stations in adjacent cells are assigned channel groups which contain completely different channels than neighbouring cells.
- Base station antennas are designed to achieve the desired coverage within a particular cell.
By limiting the coverage area within the boundaries of a cell, the same group of channels may be used to cover different cells that are separated from one another

by geographic distances large enough to keep interference levels within tolerable limits.

- The design process of selecting and allocating channel groups for all cellular base stations within a system is called frequency reuse or frequency planning.

+ Hexagonal Cell Structure:

In figure 1, cells labelled with the same letter use the same group of channels. The hexagonal cell shape is conceptual and is the simplistic model of the radio coverage for each base station. It has been universally adopted since the hexagon permits easy and manageable analysis of a cellular system. The actual radio coverage of a system is known as the footprint and is determined from old measurements and propagation prediction models. Although the real footprint is amorphous in nature, a regular cell shape is needed for systematic system design and adaptation for future growth.

If a circle is chosen to represent the coverage area of a base station, adjacent circles overlaid upon a map leave gaps or overlapping regions. A square, an equilateral triangle and a hexagon can cover the entire area without overlap and with equal area. A cell must serve the weakest mobiles typically located at the edge of the cell within the foot print. For a given distance between the center of a polygon and its farthest perimeter points, the hexagon has the largest area of the three. Thus, with hexagon, the fewest number of cell scan cover a geographic region and close approximation of a circular radiation pattern that occurs for an omni directional base antenna and free space propagation is possible.

Base station transmitters are situated either at the center of the cell (center-excited cells) or at three of the six cell vertices (edge-excited cells). Normally, omni directional antennas are used in center-excited cells and sectored directional antennas are used in edge-excited cells. Practical

system design considerations permit a base station to be positioned up to onefourth the cell radius away from the ideal location.

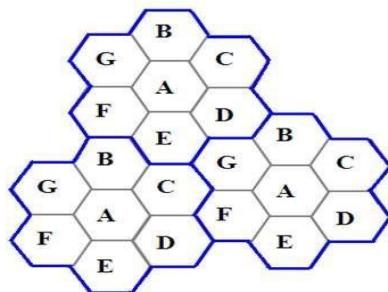
Cell Cluster:

Considering a cellular system that has a total of S duplex radio channels. If each cell is allocated a group of k channels ($k < S$)and if the S channels are divided among N cells into unique and disjoint channel groups of same number of channels, then,

$$S = kN. \quad 6.1$$

The N cells that collectively use the complete set of available frequencies is called a cluster. If a cluster is replicated M times within the system, the total number of duplex channels or capacity,

$$C = MkN= MS. \quad 6.2$$



Frequency reuse concept, Cells with the same letter use the same set of frequencies. A cell cluster is outline in blue color and replicated over the coverage area.

In this example,

The cluster size $N = 7$ and the frequency reuse factor is $1/7$ since each cell contains one-seventh of the total number of available channels.

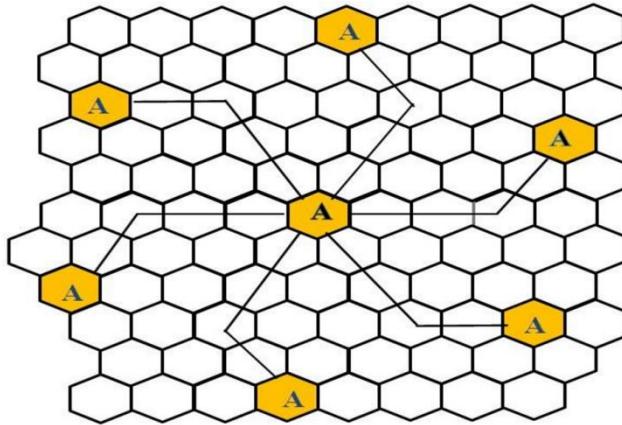
The capacity is directly proportional to M . The *factor N is called the cluster size and is typically 4, 7 or 12*. If the cluster size N is reduced while the cell size is kept constant, more clusters are required to cover a given area and hence more capacity is achieved from the

design viewpoint, the smallest possible value of N is desirable to maximize capacity over a given coverage area. The frequency reuse factor of a cellular system is $1/N$, since each cell within a cluster is assigned $1/N$ of the total available channels in the system.

Co-channel Cells:

A larger cluster size causes the ratio between the cell radius and the distance between co-channel cells to decrease reducing co-channel interference. The value of N is a function of how much interference a mobile or base station can tolerate while maintaining a sufficient quality of communications. Since each hexagonal cell has six equidistant neighbors and the line joining the centers of any cell and each of its neighbors are separated by multiples of 60 degrees, only certain cluster sizes and cell layouts are possible. To connect without gaps between adjacent cells, the geometry of hexagons is such that the number of cells per cluster, N , can only have values that satisfy,

$$N = i^2 + ij + j^2, \quad 6.3$$



Method of locating co-channel cells in a cellular system. In this figure, $N=19$ (i.e, $i = 3, j = 2$).

In this example, $N = 19$ (i.e., $i = 3, j = 2$).

Where, i and j are non-negative integers.

To find the nearest co-channel neighbours of a particular cell,

a. move i cells along any chain of hexagons then,

b. turn 60 degrees counter-clockwise and move j cells.

A.6 Steps

Follow the instructions given below to perform the experiments.

➤ Steps to perform Virtual lab Experiment:

Virtual Lab: Fading Channel and Mobile Communication

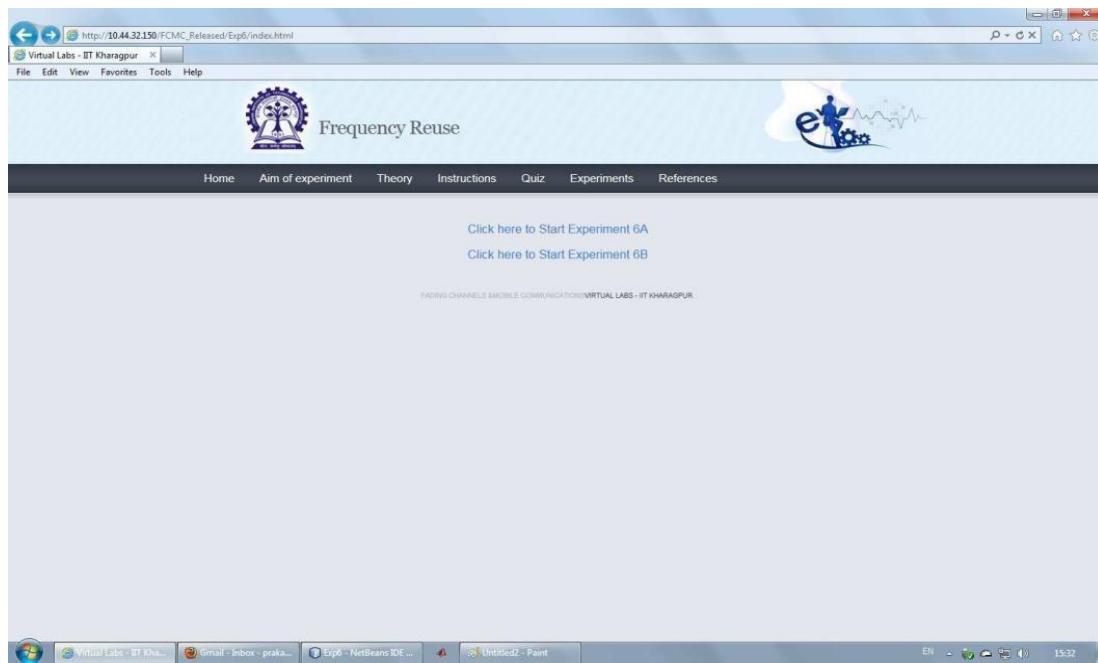
Virtual Lab Link: <http://vlabs.iitkgp.ac.in/fcmc/index.html#>

Experiment: Frequency Reuse (Co-channel cells and Cell Cluster)

Starting the Experiments:-

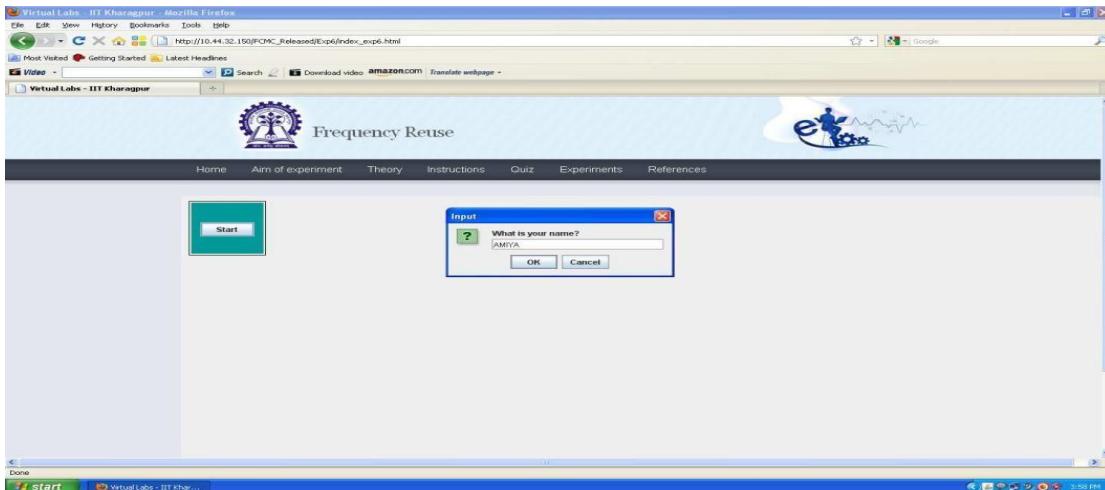
LINK: <http://vlabs.iitkgp.ernet.in/fcmc/exp6/index.html#>

- Step 1: Click on the experiment you want to do by clicking on either 'Click here to start Experiment 6A (Co-channel cell)' or 'Click here to start Experiment 6B (Cell cluster)'



Performing Experiment 1A:- •

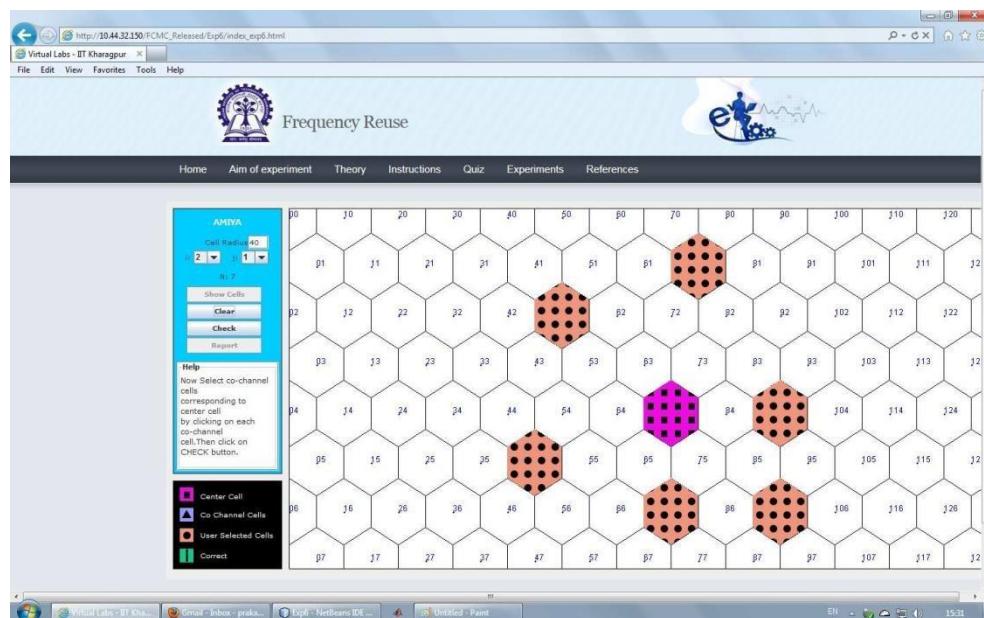
- Step 2: Let Experiment 6A (Co-channel cell) is chosen. Click on the button START. A page appears with a dialogue box asking for your name. Enter your name and click OK.



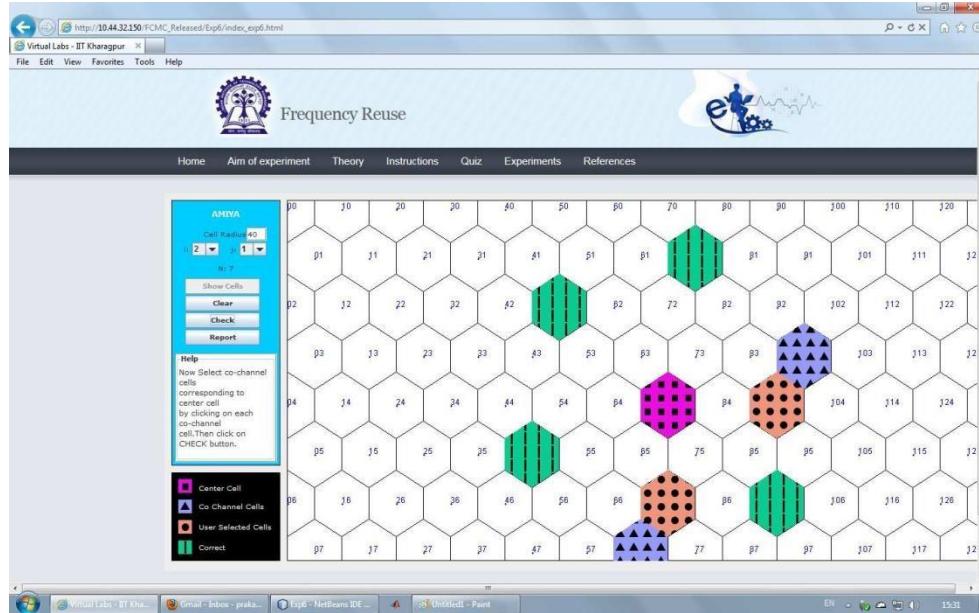
- Step 3: Choose the value of Cell Radius, i andj.
- Step 4: Click on the button Show Cells. For the given parameters, the value of Clustersize N is shown in the LHS of the page and the generated cells are shown on the RHS of the page.



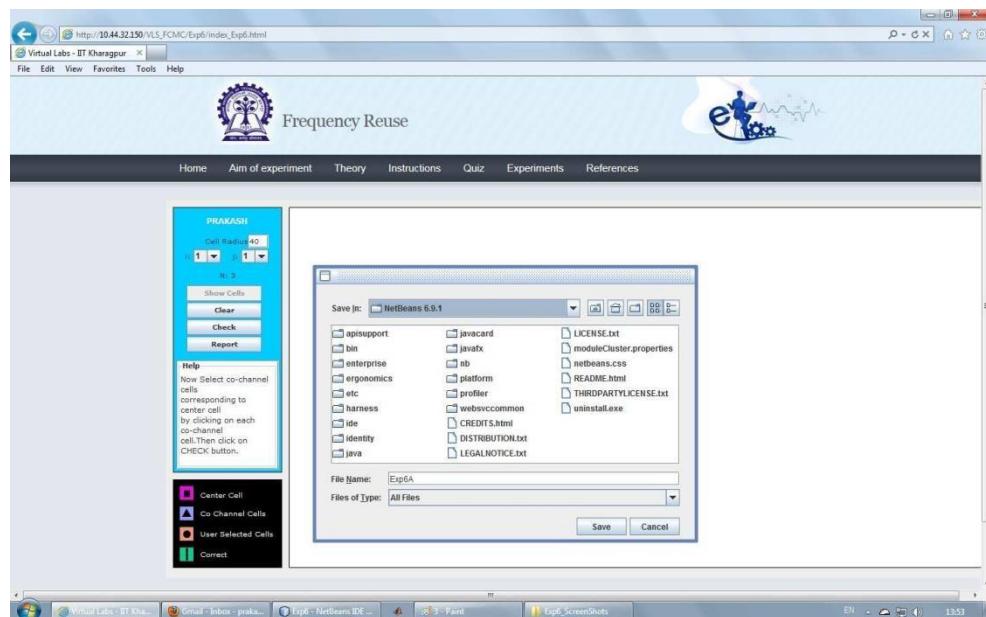
- Step 5: Within the generated cells the center cell is shown in pink colour. Select the Co-channel cells in orange colour for the center cell by finding the Co-channel cells from the formula given in the theory section.



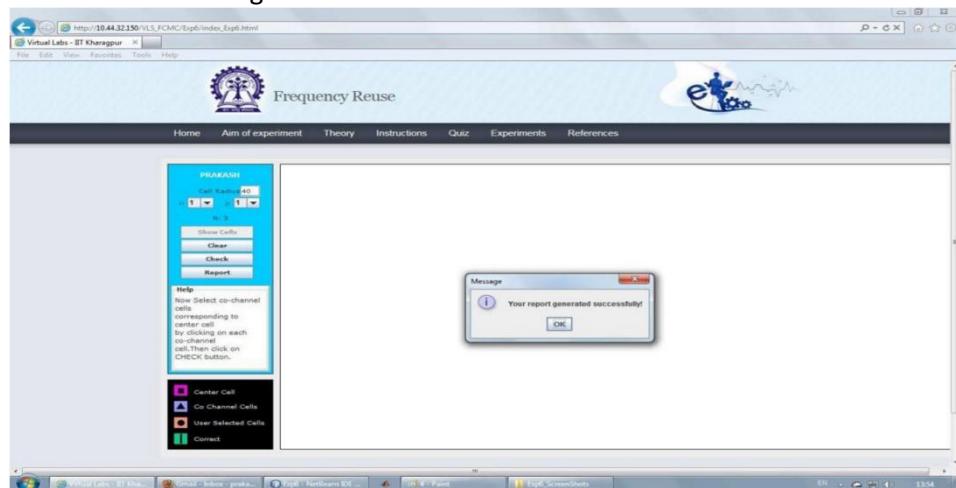
Step 6: Click on the button CHECK to see whether your manually selected Co-channel cells match with the correct Co-channel cells. If you've manually selected cells do not match with the correct Co-channel cells then the correct Co-channel cells are displayed in sky blue colour. If your manually selected Co-channel cells match with the correct Co-channel cells then the correct Co-channel cells are over-marked in green colour.



- Step 7: Click on the button REPORT to generate the report of the experiment you have performed.



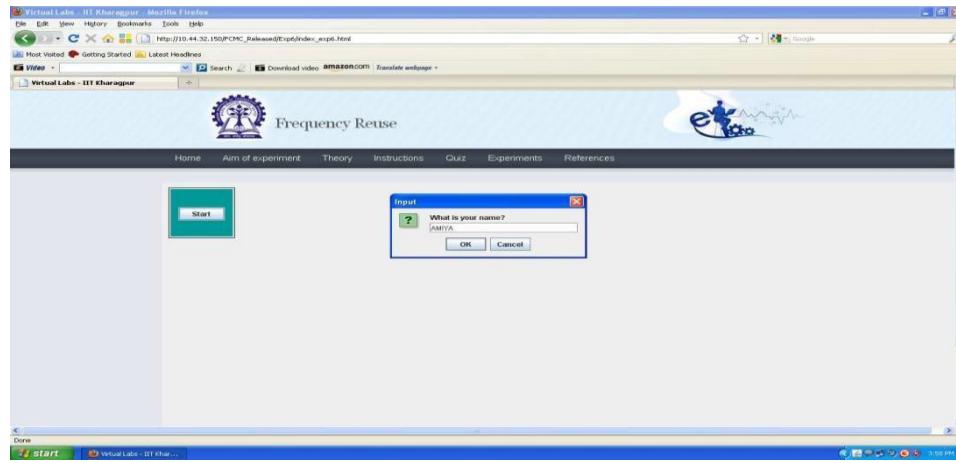
- Step 8: A dialogue box appears. Click on the button Save to save your report.
- Step 9: A dialogue box appears with the message that 'Your report has generated successfully'. Click on button OK in the dialogue box



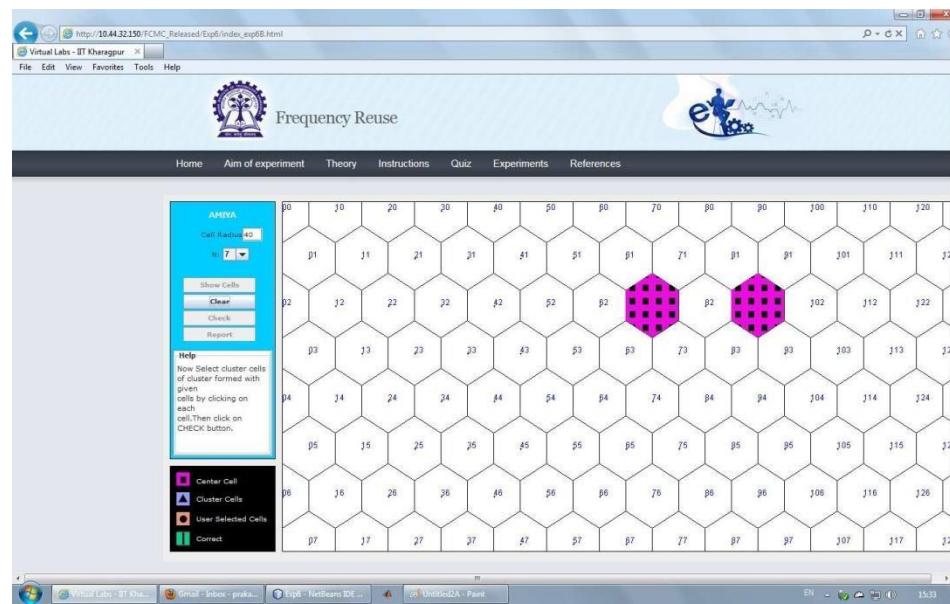
- Step 10: Now you can view the pdf report.
- Step 11: You can repeat the experiment by clicking the CLEAR button at the upper corner in the LHS of the page.

Performing Experiment 1B:-

- Step 12: Let Experiment 6B (Cell cluster) is chosen. Click on the button START. A page appears with a dialogue box asking for your name. Enter your name and click OK.



- Step 13: Choose the value of Cell Radius and Cell Cluster.
- Step 14: Click on the button Show Cells. The generated cells are shown on the RHS of the page.



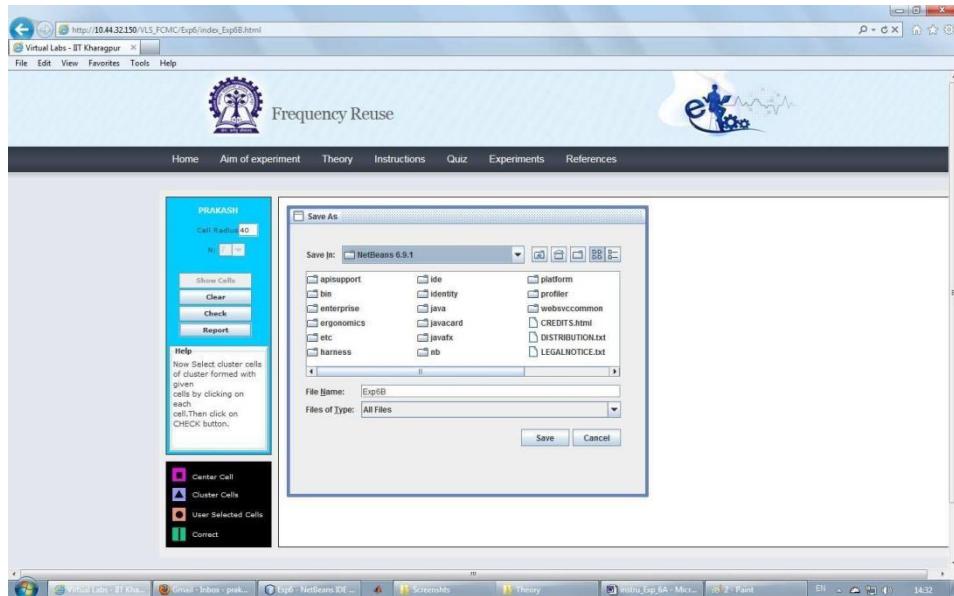
- Step 15: Within the generated cells the two extreme cells within the cell cluster is shown in pink colour. Select other cells within the cell cluster in orange colour.



- Step 16: Click on the button **CHECK** to see whether your manually selected cluster cells match with the correct cells of the cluster. If your manually selected cells do not match with the correct cells of the cluster then the correct cells of the cluster are displayed in sky blue colour. If the manually selected cells of the cluster match with the correct cells of the cluster then the correct cells of the cluster are over-marked in green colour.



- Step 17: Click on the button **REPORT** to generate the report of the experiment you have performed.



- Step 18: A dialogue box appears. Click on the button Save to save your report.
- Step 19: A dialogue box appears with the message that 'Your report has generated successfully'. Click on button OK in the dialogue box.



- Step 20: Now you can view the pdf report.
- Step 21: You can repeat the experiment by clicking the CLEAR button at the upper corner in LHS of the page.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll. No. B30	Name: Pranjal Bhatt
Class :TE COMPS B	Batch: B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student:

Co-channel.py:

```
from math import cos, sin, radians, sqrt
from tkinter import *

class Hexagon:
    def __init__(self, parent, x, y, length, color, tags):
        self.parent = parent
        self.x = x
        self.y = y
        self.length = length
        self.color = color
        self.tags = tags
        self.draw_hex()

    def draw_hex(self):
        """Draw a hexagon based on center (x, y) and edge length"""
        coords = []
        for i in range(6):
            angle_rad = radians(60 * i)
            end_x = self.x + self.length * cos(angle_rad)
            end_y = self.y + self.length * sin(angle_rad)
            coords.append((end_x, end_y))
        self.parent.create_polygon(*sum(coords, ()), fill=self.color, outline="black", tags=self.tags)

    def get_center(self):
        pass
```

```

"""Return the center coordinates of the hexagon"""
return self.x, self.y

class FrequencyReuse(Tk):
    def __init__(self, i, j, columns=16, rows=10, edge_len=30):
        super().__init__()
        self.i = i
        self.j = j
        self.N = i**2 + i*j + j**2 # Correct cluster size calculation
        self.columns = columns
        self.rows = rows
        self.edge_len = edge_len
        self.hexagons = {}
        self.first_click = True
        self.selected_hex = None
        self.co_channel_cells = []
        self.canvas = Canvas(self, width=800, height=650, bg="#Add0e1")
        self.canvas.pack()
        self.title(f"Frequency Reuse: Cluster Size {self.N}")
        self.create_grid()
        self.create_textbox()
        self.canvas.bind("<Button-1>", self.call_back)
        self.canvas.bind("<Shift-R>", self.reset_grid)
        self.mainloop()

    def create_grid(self):
        """Create a grid of hexagons"""
        size = self.edge_len
        for c in range(self.columns):
            offset = 0 if c % 2 == 0 else size * sqrt(3) / 2
            for r in range(self.rows):
                x = c * (size * 1.5) + 50
                y = (r * (size * sqrt(3))) + offset + 15
                tag = f"{r},{c}"
                self.hexagons[tag] = Hexagon(self.canvas, x, y, self.edge_len, "#fafafa", tag)

    def create_textbox(self):
        """Create the instruction text box"""
        self.textbox = Label(self.canvas, text="Select a Hexagon", font=("Helvetica", 12), bg="white", width=80)
        self.canvas.create_window(400, 600, window=self.textbox)

    def write_text(self, text):
        """Update text in instruction box"""
        self.textbox.config(text=text)

    def reset_grid(self, event=None):
        """Reset the grid and selection"""
        self.first_click = True
        self.selected_hex = None
        self.co_channel_cells = []
        self.canvas.delete("lines") # Clear previous lines
        for hexagon in self.hexagons.values():

```

```

        self.canvas.itemconfigure(hexagon.tags, fill="#fafafa")
        self.write_text("Select a Hexagon")

def find_co_channel_cells(self, start_tag):
    """Find co-channel cells based on (i, j) movement"""
    r_start, c_start = map(int, start_tag.split(","))
    co_cells = [(r_start, c_start)] # Include original cell

    # Directions to move in the hexagonal grid
    directions = [
        (self.i, self.j),
        (-self.i, -self.j),
        (self.j, -self.i-self.j),
        (-self.j, self.i+self.j),
        (self.i+self.j, -self.j),
        (-self.i-self.j, self.j)
    ]

    for dr, dc in directions:
        r_new, c_new = r_start + dr, c_start + dc
        tag_new = f"{r_new},{c_new}"
        if tag_new in self.hexagons:
            co_cells.append((r_new, c_new))

    return co_cells

def draw_line(self, tag):
    """Draw a line from the selected hexagon to the correctly guessed co-channel cell"""
    if not self.selected_hex:
        Return

    start_x, start_y = self.hexagons[self.selected_hex].get_center()
    end_x, end_y = self.hexagons[tag].get_center()
    self.canvas.create_line(start_x, start_y, end_x, end_y, fill="blue", width=2, tags="lines")

def call_back(self, evt):
    """Handle user clicks on hexagons"""
    selected_hex_id = self.canvas.find_closest(evt.x, evt.y)[0]
    selected_tag = self.canvas.gettags(selected_hex_id)[0]

    if self.first_click:
        # First selection
        self.first_click = False
        self.selected_hex = selected_tag
        self.co_channel_cells = self.find_co_channel_cells(selected_tag)

        # Highlight first selection
        self.canvas.itemconfigure(selected_tag, fill="green")
        self.write_text(f"Selected cell {selected_tag}. Now, select co-channel cells.")

    else:
        # Subsequent selections

```

```

r, c = map(int, selected_tag.split(","))
if (r, c) in self.co_channel_cells and selected_tag != self.selected_hex:
    self.canvas.itemconfigure(selected_tag, fill="green")
    self.draw_line(selected_tag) # Draw line for correct selections
    self.write_text(f"Correct! Cell {selected_tag} is a co-channel cell.")
else:
    self.canvas.itemconfigure(selected_tag, fill="red")
    self.write_text(f"Incorrect! Cell {selected_tag} is not a co-channel cell.")

if len(self.co_channel_cells) >= 7: # 1 Original + 6 Co-Cells
    self.write_text("Great! Press Shift-R to restart.")

if __name__ == '__main__':
    print("Enter i & j values. Common (i, j) values are: (1,0), (1,1), (2,0), (2,1), (3,0), (2,2)")
    i = int(input("Enter i: "))
    j = int(input("Enter j: "))
    if i == 0 and j == 0:
        raise ValueError("i & j both cannot be zero")
    elif j > i:
        raise ValueError("Value of j cannot be greater than i")
    else:
        print(f"Cluster size (N) = {i**2 + i*j + j**2}")
        FrequencyReuse(i, j)

```

Cluster.py:

```

import tkinter as tk
from tkinter import ttk
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

def hexagon(x, y, size):
    """Generate hexagon coordinates."""
    angles = np.linspace(0, 2 * np.pi, 7)
    return x + size * np.cos(angles), y + size * np.sin(angles)

def plot_clusters(cluster_size):
    """Plot the frequency reuse cell clusters with green and yellow colors."""
    fig, ax = plt.subplots(figsize=(6, 6))
    size = 10
    colors = ["green", "yellow"] # Only 2 colors for clusters

    cell_count = 1 # Start cell numbering

    for i in range(-cluster_size, cluster_size + 1):
        for j in range(-cluster_size, cluster_size + 1):
            x = i * 1.5 * size
            y = j * np.sqrt(3) * size + (i % 2) * (np.sqrt(3) / 2 * size)
            cluster_id = (i + j) % 2 # Alternating clusters
            color = colors[cluster_id]

```

```

# Draw hexagon
ax.fill(*hexagon(x, y, size), color=color, edgecolor='black')

# Display cell number at center of hexagon
text_color = "white" if color == "green" else "black"
ax.text(x, y, str(cell_count), ha='center', va='center', fontsize=10, color=text_color, fontweight='bold')

cell_count += 1 # Increment cell count

ax.set_xlim(-cluster_size * 15, cluster_size * 15)
ax.set_ylim(-cluster_size * 15, cluster_size * 15)
ax.set_aspect('equal')
ax.axis("off")
return fig

def update_plot():
    """Update the plot based on user input."""
    try:
        cluster_size = int(cluster_size_var.get())
        if cluster_size < 1:
            error_label.config(text="Please enter a positive integer")
            return
        for widget in plot_frame.winfo_children():
            widget.destroy()

        fig = plot_clusters(cluster_size)
        new_canvas = FigureCanvasTkAgg(fig, master=plot_frame)
        new_canvas.get_tk_widget().pack()
        new_canvas.draw()

        error_label.config(text="")
    except ValueError:
        error_label.config(text="Please enter a valid number")

# GUI Setup
root = tk.Tk()
root.title("Cell Clustering - Frequency Reuse")

frame = ttk.Frame(root, padding=10)
frame.pack()

# Cluster Size Input
cluster_size_var = tk.StringVar(value="3")
ttk.Label(frame, text="Cluster Size:").grid(row=0, column=0)
ttk.Entry(frame, textvariable=cluster_size_var, width=5).grid(row=0, column=1)
ttk.Button(frame, text="Update", command=update_plot).grid(row=0, column=2)

error_label = ttk.Label(frame, text="", foreground="red")
error_label.grid(row=1, column=0, columnspan=3)

```

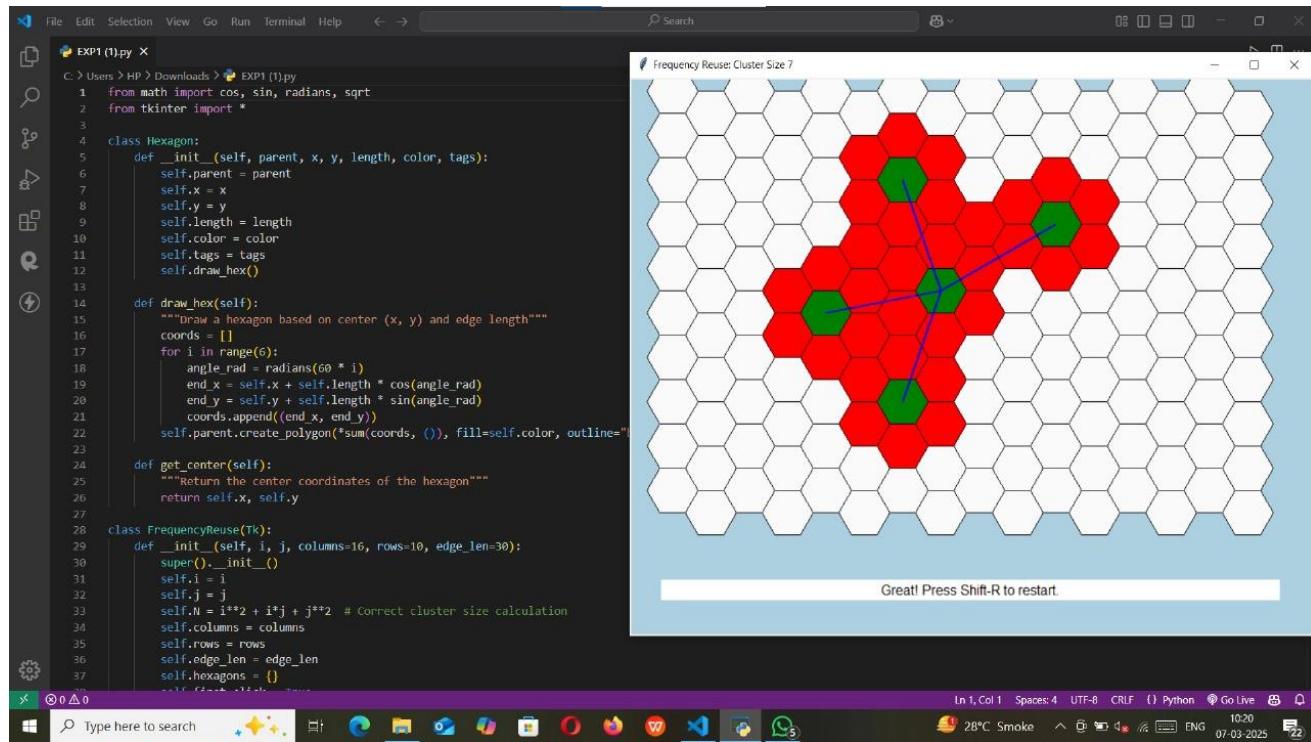
```
# Plot Frame
plot_frame = ttk.Frame(root)
plot_frame.pack()

# Initial Plot
fig = plot_clusters(int(cluster_size_var.get()))
canvas = FigureCanvasTkAgg(fig, master=plot_frame)
canvas.get_tk_widget().pack()
canvas.draw()

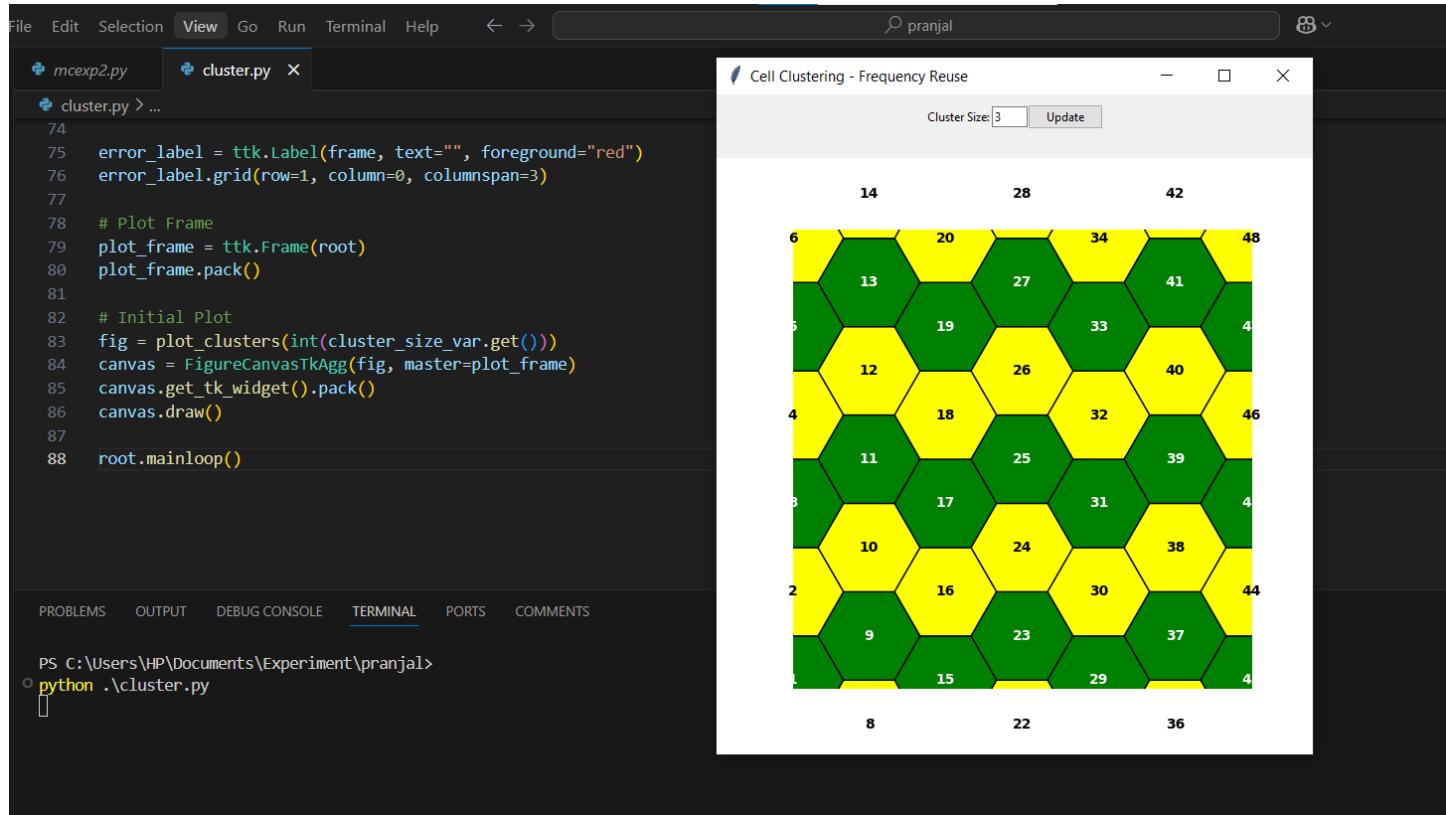
root.mainloop()
```

B.2 Input and Output:

Co-channel:



Cluster:



```

File Edit Selection View Go Run Terminal Help ← → ⚙ pranjal
mexp2.py cluster.py ✘
cluster.py > ...
74
75     error_label = ttk.Label(frame, text="", foreground="red")
76     error_label.grid(row=1, column=0, columnspan=3)
77
78     # Plot Frame
79     plot_frame = ttk.Frame(root)
80     plot_frame.pack()
81
82     # Initial Plot
83     fig = plt.subplots(int(cluster_size_var.get()))
84     canvas = FigureCanvasTkAgg(fig, master=plot_frame)
85     canvas.get_tk_widget().pack()
86     canvas.draw()
87
88 root.mainloop()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\HP\Documents\Experiment\pranjal>
python .\cluster.py

B.3 Observations and learning:

Co-channel Cells Identification:

- Cells using the same frequency were identified based on the frequency reuse factor ($1/N_1/N_1/N$), where N_{NN} is the cluster size.
- The co-channel cells were separated by a sufficient distance to minimize interference.
- Example: For a cluster size $N=7N = 7N=7$, co-channel cells formed a repeating pattern, maintaining adequate spacing.

Cluster Formation:

- The entire geographic area was divided into clusters of cells.
- Each cluster used all available frequencies ($S=kNS = kNS=kN$) without overlap between adjacent clusters.

Hexagonal Cell Structure:

1. The hexagonal layout was observed to provide efficient coverage without gaps or overlaps.
2. Co-channel cells were located using the formula $N=i^2+ij+j^2N = i^2 + ij + j^2N = i^2+ij+j^2$, ensuring equal spacing between them.

Frequency Reuse Efficiency:

1. Reuse of frequencies increased the system's capacity to accommodate more users while limiting interference.
2. Smaller cluster sizes ($N=4N = 4N=4$ or $N=7N = 7N=7$) allowed higher frequency reuse but required careful planning to control interference.

Impact of Cluster Size:

1. Larger cluster sizes reduced co-channel interference but also decreased capacity.

Smaller cluster sizes provided higher capacity but required tighter interference management.

B.4 Conclusion:

The experiment demonstrates that frequency reuse is an essential concept for efficient spectrum utilization in cellular networks. By allocating the same frequency to co-channel cells separated by sufficient distance, interference is minimized while maximizing user capacity. The hexagonal cell structure ensures optimal coverage without overlaps or gaps, making it ideal for network design. Smaller cluster sizes increase capacity but require careful interference management. Overall, frequency reuse enables scalable, high-capacity cellular systems, forming the foundation of modern communication technologies like GSM, LTE, and 5G.

B.5 Question of Curiosity

1. With Example explain Co-channel cells

Definition:

Co-channel cells are cells in a cellular network that use the same frequency for communication but are separated by a sufficient distance to prevent interference. These cells are part of the frequency reuse concept, which allows the reuse of limited frequencies in non-adjacent cells to increase system capacity.

Example:

Suppose a cellular system has a total of 21 frequencies, and each cell in a cluster of $N=7$ uses 3 frequencies.

These 7 cells form a cluster, and frequencies are assigned as:

Cell A: Frequencies (f_1, f_2, f_3)

Cell B: Frequencies (f_4, f_5, f_6)

Cell C: Frequencies (f_7, f_8, f_9) , and so on.

Now, the same frequencies (f_1, f_2, f_3) can be reused in another cell, labeled A', which is geographically distant from Cell A. A' is a co-channel cell to A.

To locate co-channel cells geometrically:

Use the formula: $N = i^2 + ij + j^2$ where i and j are integers.

For $N=7$: $i=2, j=1$.

Start at Cell A.

Move 2 cells in a straight line.

Turn 60° counterclockwise and move 1 cell to find the nearest co-channel cell.

2. Define following Term: a) cell b) Frequency Reuse c) Cell Splitting

a) Cell

A cell is a small geographical area in a cellular network covered by a base station using a specific set of frequencies.

Each cell serves users within its boundary and ensures efficient use of spectrum by allowing frequency reuse in other cells.

Cells are typically represented as hexagons for coverage analysis.

b) Frequency Reuse

Frequency reuse is the technique of using the same frequency bands in geographically separated cells within a cellular network. This allows efficient spectrum utilization and supports more users in a limited frequency spectrum.

Example: If Cell A uses frequency f_1 , another cell (A') separated by sufficient distance can also use f_1 without causing interference.

c) Cell Splitting

Cell splitting is a technique used to increase the capacity of a cellular network by dividing a large cell into smaller cells. Each smaller cell has its own base station and operates at a lower power level.

Purpose: Accommodate more users in high-density areas by increasing the number of cells.

Example: A cell with a radius of 10 km can be split into 4 cells with a radius of 5 km each, doubling the capacity in that area.

Experiment No.2: CDMA

PART A

(PART A: TO BE REFERRED BY STUDENTS)

A.1 Aim: To implement a basic function of Code Division Multiple Access (CDMA) to test the orthogonality and autocorrelation of a code to be used for CDMA operation. Write an application based on the above concept.

A.2 Prerequisite: Knowledge of multiplexing schemes

A.3 Objectives: To understand the importance of security in wireless communication by means of spread spectrum technologies.

A.4 Outcomes: Student will be able to articulate the knowledge of GSM, CDMA & Bluetooth technologies and demonstrate it.(LO-2)

A.5 Tools Used/programming language: Java

A.6 Theory:

- **Code-division multiple access (CDMA)** is a channel access method **used by various radio communication technologies**. CDMA is an example of multiple accesses, where several transmitters can send information simultaneously over a single communication channel. This allows several users to share a band of frequencies (see bandwidth). To permit this without undue interference between the users, CDMA employs spread spectrum technology and a special coding's scheme (where each transmitter is assigned a code).

CDMA issued as the access method in many mobile phone standards. IS-95, also called "cdma One", and its 3G evolution CDMA2000, are often simply referred to as "CDMA", but UMTS, the 3G standard used by GSM carriers, also uses "wideband CDMA", or W-CDMA, as well as TD-SCDMA and TD-SCDMA, as its radio technologies.

▪ **CDMA Orthogonality:**

Techniques generally used are direct sequence spread spectrum modulation (DS-CDMA), frequency hopping or mixed CDMA detection (JD-CDMA). Here, a signal is generated which extends over a wide bandwidth. A code called spreading code is used to perform this action. Using a group of codes, which are orthogonal to each other, it is possible to select a signal with a given code in the presence of many other signals with different orthogonal codes.

▪ **CDMA Autocorrelation:**

Autocorrelation of the sequence, it determines the ability to synchronize and lock the spreading code for the received signal.

A.7 Procedure:

- ✓ The station encodes its data bit as follows.
 - If bit = 1 then +1
 - If bit = 0 then -1
 - no signal(interpreted as 0) if station is idle

- ✓ Each station is allocated a different orthogonal sequence (code) which is N bit long for N stations

- ✓ Each station does a scalar multiplication of its encoded data bit and code sequence.
- ✓ The resulting sequence is then stored on the channel.
- ✓ Since the channel is common, amplitudes add up and hence resultant channel sequence is the sum of sequences from all channels.

- ✓ If station 1 wants to listen to station 2, it multiplies (inner product) the channel sequence with code of station S2
- ✓ The inner product is then divided by N to get data bit transmitted from station 2.

■ **Working**

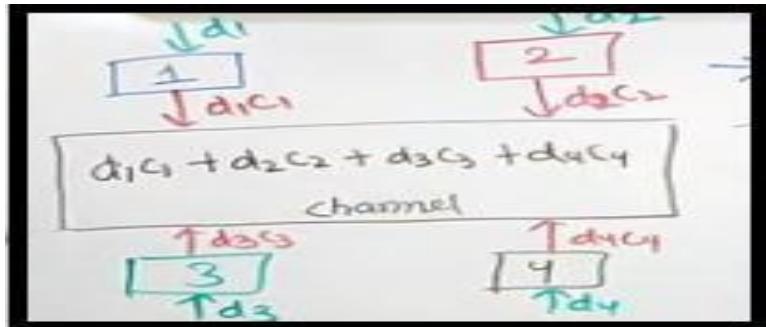
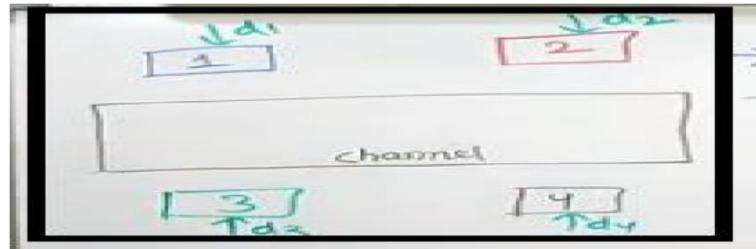
- ✓ CDMA uses orthogonal codes to transmit different signal over the same channel
- ✓ CDMA is used in 3'rd generation wireless communication like CDMA 2000, w-CDMA,HSDPA (high speed downlink packet access), HSUPA(high speed uplink packet access) ✓

CDMA stands for Code Division Multiple Access.

- ✓ It is a digital cellular standard that utilizes spread-Spectrum Technology.
- ✓ It spreads the signal over a fully available spectrum or over multiple channels through division.

- ✓ It is a more secure and private line.
- ✓ It has good voice and data communication capabilities.
- ✓ The information is sent simultaneously through several transmitters over a single communication channel.

- ✓ **Consider there is single channel having four users, user1,user2,user3,user4**

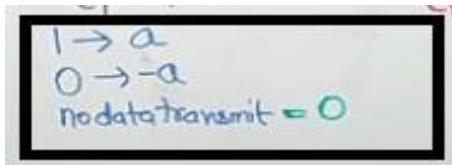


- ✓ Assume there are four orthogonal codes

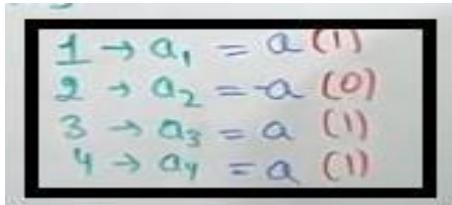
$$\begin{array}{l}
 C_1, C_2, C_3, C_4 \rightarrow \text{orthogonal code} \\
 C_1 \rightarrow 1 \ 1 \ 1 \ 1 \quad C_1 C_2 = 1 + (-1) + 1 + (-1) = 0 \\
 C_2 \rightarrow 1 \ -1 \ 1 \ -1 \\
 C_3 \rightarrow 1 \ 1 \ -1 \ -1 \\
 C_4 \rightarrow 1 \ -1 \ -1 \ 1
 \end{array}$$

$$\begin{array}{l}
 C_1, C_2, C_3, C_4 \rightarrow \text{orthogonal code} \\
 C_1 \rightarrow 1 \ 1 \ 1 \ 1 \quad C_1 C_2 = 1 + (-1) + 1 + (-1) = 0 \\
 C_2 \rightarrow 1 \ -1 \ 1 \ -1 \quad C_2 C_4 = 1 + 1 - 1 - 1 = 0 \\
 C_3 \rightarrow 1 \ 1 \ -1 \ -1 \quad C_3 \times C_3 = (+) + (-) + (+) = 4 \\
 C_4 \rightarrow 1 \ -1 \ -1 \ 1
 \end{array}$$

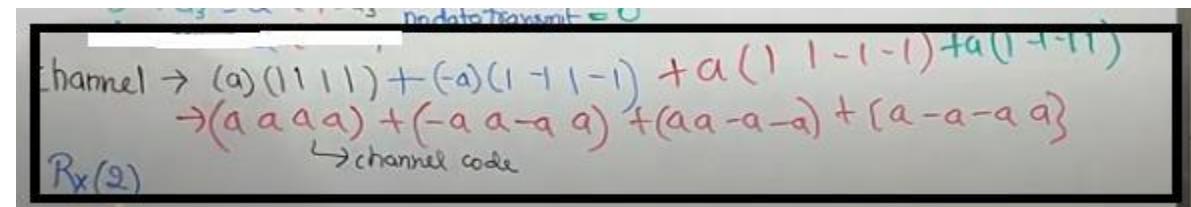
- ✓ If two diff orthogonal codes are multiplied it will always give 0; this is the property of orthogonal code.
- ✓ User send data 1,0 (stream of data) or user may be silent that is no data to transmit.



- ✓ Users are having following data



- ✓ Data in channel is;



Handwritten notes:

$$\text{channel} \rightarrow (a)(1111) + (-a)(1-11-1) + a(11-1-1) + a(1-11)$$

$$\rightarrow (aaaa) + (-aaa) + (aa-aa) + (a-a-aa)$$

$\xrightarrow{\text{channel code}}$

$$Rx(2)$$

- ✓ If data 2 want to receive then

$$R2 = C(x) * C2$$

$$\rightarrow (aaaa + -aaa + aa-aa + a-a-aa) \{ 1-1-1 \}$$

$$\rightarrow a+a-a+a-a-a+a+d+d+d+d$$

$$\rightarrow -4a$$

And hence we are getting $-a$ after dividing by 4 and data send by user 2 is $-a$

Sample Output:

The screenshot shows a Windows command prompt window titled 'C:\Windows\System32\cmd.exe'. The window contains the following text:

```
Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. All rights reserved.

E:\Course_Subjects\FH 2022\MC_TE_B_FH 2022\MC_Experiments\Python>python CDMA.py
Enter the data bits :
Enter D1 :12
Enter D2 :13
Enter D3 :14
Enter D4 :15
Resultant Channel [54 -2 -4  0]
Enter the station to listen for C1=1 ,C2=2, C3=3 C4=4 : 3
Inner Product [54 -2  4  0]
Data bit that was sent 14.0

E:\Course_Subjects\FH 2022\MC_TE_B_FH 2022\MC_Experiments\Python>
```

The taskbar at the bottom of the screen includes icons for File Explorer, Google Chrome, Task View, and others. The system tray shows the date and time as 21-01-2022 17:11:32.

PART B**(PART B: TO BE COMPLETED BY STUDENTS)**

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case there is no ERP access available)

Roll No. B30	Name: Pranjal Bhatt
Class : TE COMPS B	Batch :B2
Date of Experiment:	Date of Submission
Grade :	

B.1 Question of Curiosity:

Q.1: Source Code (students need to implement CDMA using any programming language like Java, Python , etc)

```
import numpy as np

c1 = [1, 1, 1, 1]
c2 = [1, -1, 1, -1]
c3 = [1, 1, -1, -1]
c4 = [1, -1, -1, 1]
rc = []

print("Enter the data bits:")
d1 = int(input("Enter D1: "))
d2 = int(input("Enter D2: "))
d3 = int(input("Enter D3: "))
d4 = int(input("Enter D4: "))

r1 = np.multiply(c1, d1)
r2 = np.multiply(c2, d2)
```

```
r3 = np.multiply(c3, d3)
r4 = np.multiply(c4, d4)

resultant_channel = r1 + r2 + r3 + r4
print("Resultant Channel:", resultant_channel)

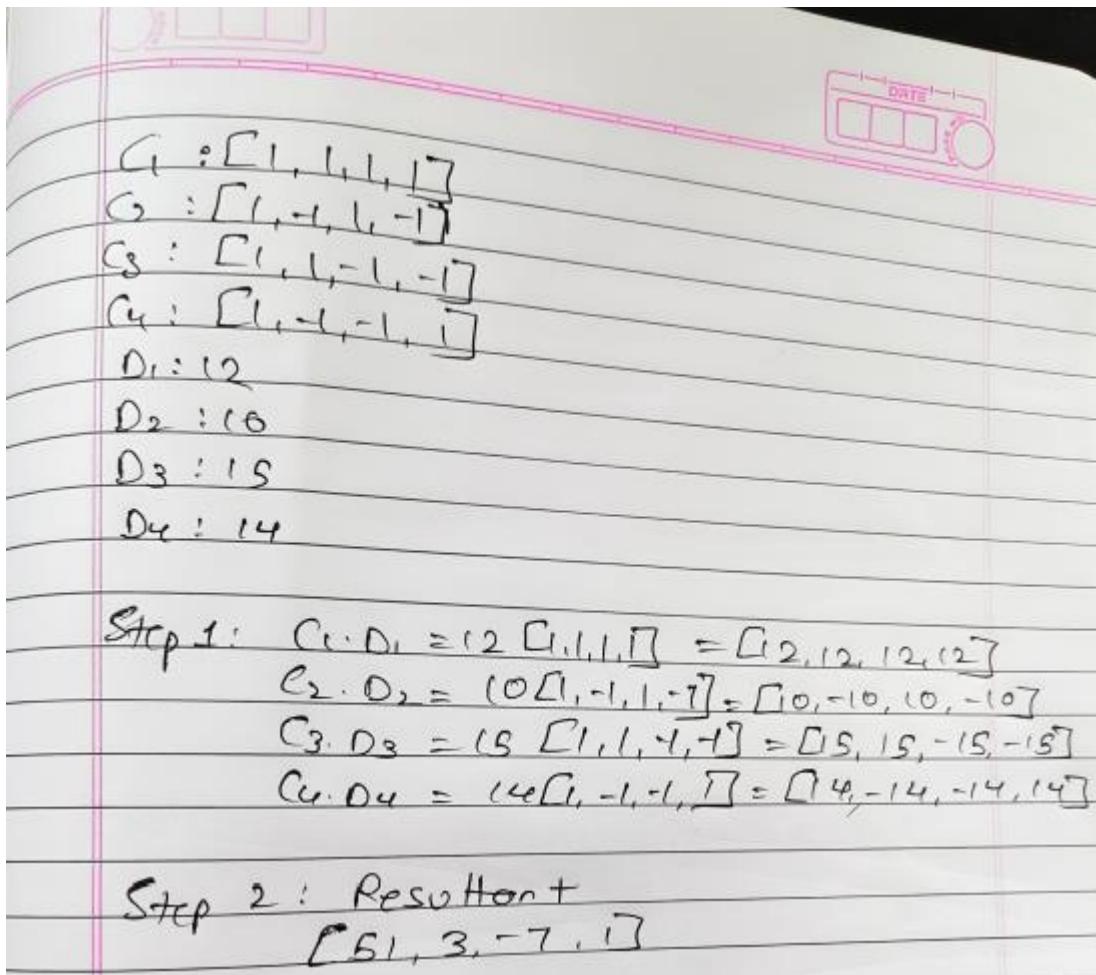
Channel = int(input("Enter the station to listen for (C1=1, C2=2, C3=3, C4=4): "))
if Channel == 1:
    rc = c1
elif Channel == 2:
    rc = c2
elif Channel == 3:
    rc = c3
elif Channel == 4:
    rc = c4

inner_product = np.multiply(resultant_channel, rc)
print("Inner Product:", inner_product)

res1 = sum(inner_product)
data = res1 / len(inner_product)
print("Data bit that was sent:", data)
```

Q.2: Output of CDMA

```
PS C:\Users\HP\Documents\Experiment\pranjal> python .\mcexp2.py
Enter the data bits:
Enter D1: 112
Enter D2: 10
Enter D3: 15
Enter D4: 14
Resultant Channel: [151 103 93 101]
Enter the station to listen for (C1=1, C2=2, C3=3, C4=4): 3
Inner Product: [ 151  103 -93 -101]
Data bit that was sent: 15.0
○ PS C:\Users\HP\Documents\Experiment\pranjal>
```



Handwritten notes on a lined notebook page:

- $G_1 : [1, 1, 1, 1]$
- $G_2 : [1, -1, 1, -1]$
- $G_3 : [1, 1, -1, -1]$
- $G_4 : [1, -1, -1, 1]$
- $D_1 : 12$
- $D_2 : 16$
- $D_3 : 15$
- $D_4 : 14$

Step 1:

$$C_1 \cdot D_1 = 12 [1, 1, 1, 1] = [12, 12, 12, 12]$$

$$C_2 \cdot D_2 = 10 [1, -1, 1, -1] = [10, -10, 10, -10]$$

$$C_3 \cdot D_3 = 15 [1, 1, -1, -1] = [15, 15, -15, -15]$$

$$C_4 \cdot D_4 = 14 [1, -1, -1, 1] = [14, -14, -14, 14]$$

Step 2: Resultant

$$[51, 3, -7, 17]$$

Q.3: Explain CDMA with one suitable example.

CDMA (Code Division Multiple Access) is a communication technique that allows multiple signals to occupy a single transmission channel, optimizing the use of available bandwidth. Each communication is assigned a unique code, allowing multiple users to transmit over the same frequency without interfering with each other.

How CDMA works:

- Unique Codes:** Each user is assigned a unique code, often called a **spread spectrum code**. This code is used to "spread" the signal over a wide frequency band.
- Signal Transmission:** When users transmit their data, the information is multiplied by their unique code. This results in a signal that is spread out across the spectrum.
- Reception:** At the receiver end, the signal is decoded using the same unique code, which extracts the original information from the spread signal.

Since each user's signal is spread differently using their unique code, multiple users can share the same frequency channel without interfering with each other.

Example of CDMA:

Imagine a situation in a city where many people are trying to make calls using the same frequency channel. Without CDMA, all signals would overlap, causing interference. However, with CDMA, each user is assigned a unique code.

For instance:

- **User A** is assigned the code "101010"
- **User B** is assigned the code "110011"
- **User C** is assigned the code "111000"

Now, when each user transmits their signal, it is multiplied by their unique code. So, even though all users are transmitting at the same time over the same frequency, their signals don't interfere with each other because each signal is spread differently due to the unique codes.

At the receiver end, the signals are decoded by applying the same codes used for transmission, which separates the signals of different users and ensures they are correctly received.

Q.4: What is spread spectrum? List types of spread spectrum technologies.

Spread Spectrum is a technique used in communication systems to spread the signal over a wider frequency band than the minimum required bandwidth. This is done to improve the performance and security of the communication system. The spread spectrum method helps reduce interference, improve signal-to-noise ratio (SNR), and make the system more resistant to jamming and eavesdropping.

Types of Spread Spectrum Technologies:

1. Frequency Hopping Spread Spectrum (FHSS):

- **How it works:** The transmitter rapidly changes its carrier frequency in a sequence known to both the transmitter and receiver. This hopping occurs over a wide band of frequencies.
- **Example:** Bluetooth devices use FHSS to minimize interference and improve security by constantly switching frequencies within a set range.

1. Direct Sequence Spread Spectrum (DSSS):

- How it works:** The data signal is multiplied by a higher-frequency pseudo-random noise (PN) code, spreading the signal over a wider frequency band. Both the transmitter and receiver must use the same code to decode the signal.
- Example:** Wi-Fi (802.11b/g) uses DSSS to transmit data over a wide frequency spectrum, which reduces the likelihood of interference and increases the reliability of the connection.

Q.5: Differentiate between FDM and FHSS.

Feature	Frequency Division Multiplexing (FDM)	Frequency Hopping Spread Spectrum (FHSS)
Definition	A multiplexing technique that divides the available bandwidth into multiple frequency channels for simultaneous transmission.	A spread spectrum technique that rapidly switches (hops) between different frequencies during transmission.
Primary Purpose	Used to transmit multiple signals simultaneously over different frequency bands.	Used to enhance security and resistance to interference by spreading the signal over multiple frequencies.
Bandwidth Usage	Fixed frequency allocation per channel.	Uses a wide range of frequencies but transmits at a single frequency at any given time.
Interference Handling	Less resistant to interference; affected by narrowband interference.	Highly resistant to interference and jamming due to frequency hopping.
Security	Less secure as frequencies remain fixed.	More secure as hopping makes eavesdropping difficult.
Synchronization	Requires precise frequency allocation and synchronization.	Requires synchronization for hopping sequence but is more robust against signal interception.

Feature	Frequency Division Multiplexing (FDM)	Frequency Hopping Spread Spectrum (FHSS)
Example Applications	FM radio, television broadcasting, cable TV, and telephone networks.	Bluetooth, military communications, and some wireless LANs.

B.2 Conclusion:

This experiment demonstrates the concept of **Code Division Multiplexing (CDM)** using orthogonal codes for data transmission and retrieval. By assigning unique orthogonal codes to different data streams, multiple signals are transmitted simultaneously over a shared channel without interference.

The **resultant channel** is obtained by summing the encoded signals, and the original data bit is retrieved by performing an inner product operation with the corresponding code. This method ensures efficient multiplexing and accurate data recovery, provided that the codes are truly orthogonal.

Such a technique is fundamental in **spread spectrum communication**, particularly in applications like **CDMA (Code Division Multiple Access)**, where multiple users can share the same frequency bandwidth without causing interference.

Experiment No 3: GSM Security

PART A

(PART A: TO BE REFERRED BY STUDENTS)

A.1 Aim: To implement GSM security algorithms (A3/A5/A8)

A.2 Objectives: To understand the security algorithms in mobile networks

A.3 Outcomes: Student will be able to implement security algorithms for mobile communication network. (LO-4)

A.4 Tools Used/programming language: Java, Python etc

A.5 Theory:

- Authentication verifies identity and validity of SIM card to the network and ensures that subscriber has access to the network.
- Term used
 - ✓ Ki= **individual subscriber authentication key**, it is 32 bit number and present only in SIM card and stored in authentication center.
 - ✓ RAND= **random 128 bit number generated by AUC** (authentication center) when network request to authenticate the subscribers.

- ✓ SRES (signed responses) = 32 bit crypto variable used in authentication process.
- ✓ Kc = 64 bit cipher key.
- MS is challenged by given RAND by the network.

▪ Security in GSM

- Three algorithms have been specified to provide security services in GSM. **Algorithm A3** is used for **authentication**, **A5** for **encryption**, and **A8** for the **generation of a cipher key**.
- In the GSM standard **only algorithm A5 was publicly available, whereas A3 and A8 were secret**, but standardized with open interfaces.
- **Network providers can use stronger algorithms for authentication** – or users can apply stronger end-to-end encryption.
- **Algorithms A3 and A8 (or their replacements)** are located on the **SIM** and in the **AUC** and can be proprietary.
- **Only A5 which is implemented in the devices** has to be identical for all providers.

⊕ Subscriber Authentication

For subscriber authentication algorithm used is A3

1. A3 algorithm is inbuilt inside SIM and AUC, Input for A3 is Ki and RAND
2. Ki=Stored inside SIM(**kiis encrypted inside SIM card**) and not share on network and also present in AUC of MSC.
3. Before a subscriber can use any service from the **GSM network**, he or she must be authenticated. **Authentication is based on the SIM**, which stores the individual **authentication key Ki**, the **user identification IMSI**, and the algorithm used for authentication **A3**.
4. When user want to access **GSM network** IMSI number from SIM send to **MSC** then **HLR** then to **AUC**.

5. Now AUC check **IMSI number** is present or not and identify associated **Ki value** (**Ki** is fixed), in this procedure AUC generate **RAND number** which is different for every new user request.
6. AUC using authentication algorithm A3 (input to A3 are **ki** and **RAND**) calculate SRES as output of A3 and AUC using algorithm A8 of cipher generation (input to A8 are **ki** and **RAND**) calculate **Kc** and send these SRES, **Kc** and **RAND** to HLR then from HLR to MSC. These three terms SRES, **Kc** and **RAND** are called as triplet.
7. MSC now send only **RAND** value to MS
8. MS using algorithm A3 (input to A3 is **Ki** and **RAND**) calculate SRES and using algorithm A8 calculate **Kc** and send these SRES and **Kc** to MSC
9. MSC check SRES receive from MS and Network are same or not. If both are same user is authenticated and connection is set up.

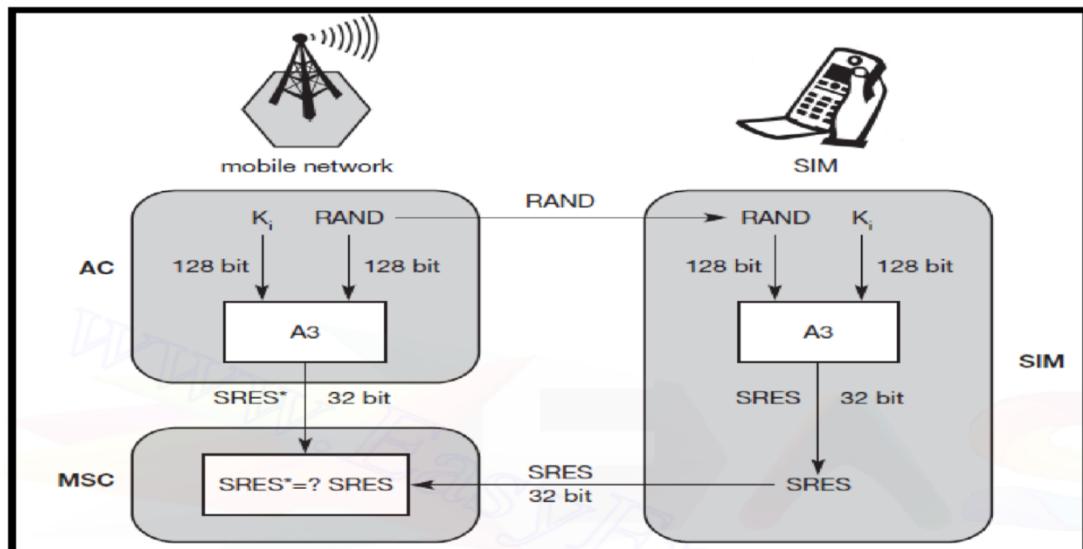


Figure: Subscriber Authentication

Encryption

1. To ensure privacy, all messages containing user-related information are encrypted in GSM over the air interface.
2. After authentication, MS and BSS can start using encryption by applying the cipher key **Kc**

3. **Kc** is generated using the individual key K_i and a random value by applying the algorithm A8. Note that the SIM in the MS and the network both calculate the same K_c based on the random value RAND. **The key K_c itself is not transmitted over the air interface.**
4. **MS and BTS can now encrypt and decrypt data using the algorithm A5 and the cipher key K_c .** As Figure shows, K_c should be a 64 bit key – which is not very strong, but is at least a good protection against simple eavesdropping. However, the publication of A3 and A8 on the internet showed that in certain implementations 10 of the 64 bits are always set to 0, so that the real length of the key is thus only 54 consequently, the encryption is much weaker.
5. **Note:** An **eavesdropping attack**, also known as a sniffing or snooping **attack**, is a theft of information as it is transmitted over a network by a computer, smart-phone, or another connected device. The **attack** takes advantage of unsecured network communications to access data as it is being sent or received by its user. **Eavesdropping** is the act of intercepting communications between two points.

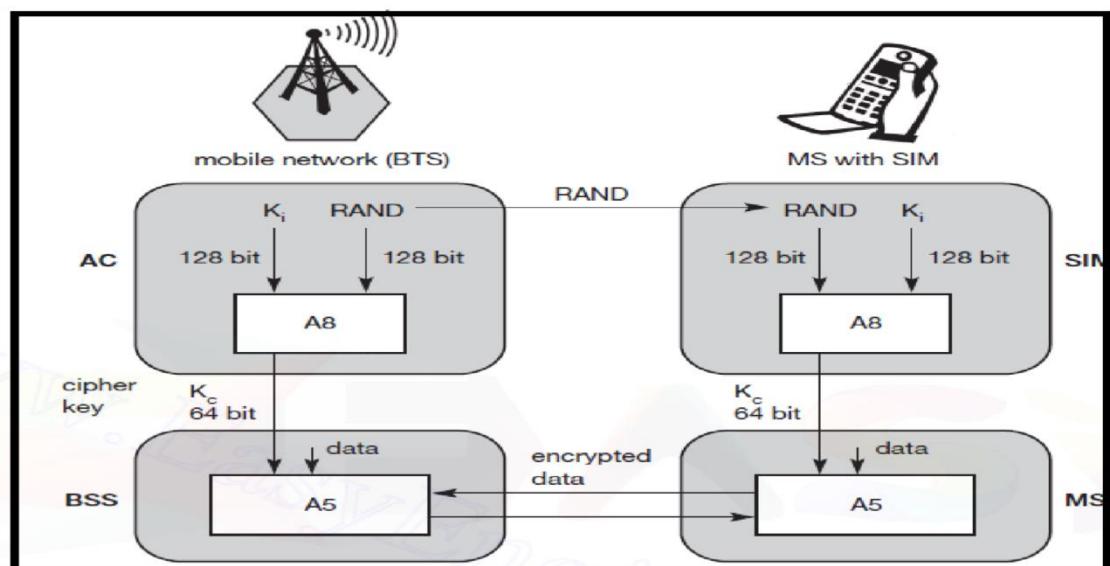


Figure: Data Encryption

A.6 Sample Source Code:

<https://www.theprogrammingcodeswarehouse.com/2020/04/implementation-of-a-3-security.html>

```
import random
k=random.getrandbits(128)
m=random.getrandbits(128) kb=bin(k)[2:]
mb=bin(m)[2:] kbl=kb[0:64] kbr=kb[64:]
mbl=mb[0:64] mbr=mb[64:]
a1=int(kbl,2)^int(mbr,2)
a2=int(kbr,2)^int(mbl,2) a3=a1^a2
a4=bin(a3)[2:].zfill(64) a5=a4[0:32]
a6=a4[32:] a7=int(a5,2)^int(a6,2)
print("128 Bit Key = ",kb)
print("128 Random Bits Generated = ",mb) print("RES/SRES
= ",bin(a7)[2:].zfill(len(a5)))
```

A.6 Sample Output:

128 Bit Key
=1111101110100110010000010010011000100111001111010011101011010001111000111000001
111

011101110110111010100010110101000111010001

128 Random Bits Generated
=11000001000100010110001011100100110110011001000110101110001001000010100101
001

000001001111000000100001100100111111000100

RES/SRES=1111011011010000001011110001101

PART B**(PART B: TO BE COMPLETED BY STUDENTS)**

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case there is no ERP access available)

Roll No. B30	Name: Pranjal Bhatt
Class : TE COMPS B	Batch :B2
Date of Experiment:	Date of Submission
Grade :	

B.1 Question of Curiosity:

Q.1: Source Code (students need to implement GSM Security Algorithm using any programming language like Java, Python, etc)

```
import re
import copy
import sys
import random

def generate_binary_128():
    return bin(random.getrandbits(128))[2:].zfill(128)

# Generate random values
k = generate_binary_128()
m = generate_binary_128()

kbl, kbr = k[:64], k[64:]
mbl, mbr = m[:64], m[64:]
```

```

a1, a2 = int(kbl, 2), int(kbr, 2)
a3 = a1 ^ a2

a4 = bin(a3)[2:].zfill(64)
a5, a6 = a4[:32], a4[32:]
a7 = int(a5, 2)

print("128 Bit Key =", k)
print("128 Random Bits Generated =", m)
print("RES/SRES =", bin(a7)[2:].zfill(len(a5)))

# Register lengths
reg_x_length, reg_y_length, reg_z_length = 19, 22, 23

key_one = ""
reg_x, reg_y, reg_z = [], [], []

def loading_registers(key):
    global reg_x, reg_y, reg_z
    reg_x, reg_y, reg_z = [int(bit) for bit in key[:reg_x_length]], [int(bit) for bit in
key[reg_x_length:reg_x_length+reg_y_length]], [int(bit) for bit in key[reg_x_length+reg_y_length:]]]

def set_key(key):
    global key_one
    if len(key) == 64 and re.match("^(01)+$", key):
        key_one = key
        loading_registers(key)
        return True
    return False

def get_majority(x, y, z):
    return 1 if (x + y + z) > 1 else 0

def get_keystream(length):
    reg_x_temp, reg_y_temp, reg_z_temp = reg_x[:], reg_y[:], reg_z[:]
    keystream = []

    for _ in range(length):
        majority = get_majority(reg_x_temp[8], reg_y_temp[10], reg_z_temp[10])
        if reg_x_temp[8] == majority:
            reg_x_temp.insert(0, reg_x_temp[13] ^ reg_x_temp[16] ^ reg_x_temp[17] ^ reg_x_temp[18])
            reg_x_temp.pop()
        if reg_y_temp[10] == majority:
            reg_y_temp.insert(0, reg_y_temp[20] ^ reg_y_temp[21])

```

```

    reg_y_temp.pop()
if reg_z_temp[10] == majority:
    reg_z_temp.insert(0, reg_z_temp[7] ^ reg_z_temp[20] ^ reg_z_temp[21] ^ reg_z_temp[22])
    reg_z_temp.pop()
keystream.append(reg_x_temp[18] ^ reg_y_temp[21] ^ reg_z_temp[22])

return keystream

def to_binary(plain):
    return ''.join(format(ord(x), '08b') for x in plain)

def convert_binary_to_str(binary):
    return ''.join(chr(int(binary[i:i+8], 2)) for i in range(0, len(binary), 8))

def encrypt(plain):
    binary = to_binary(plain)
    keystream = get_keystream(len(binary))
    return ''.join(str(int(binary[i]) ^ keystream[i]) for i in range(len(binary)))

def decrypt(cipher):
    keystream = get_keystream(len(cipher))
    binary = ''.join(str(int(cipher[i]) ^ keystream[i]) for i in range(len(cipher)))
    return convert_binary_to_str(binary)

def main_fun():
    key = kbl
    set_key(key)
    print("Using generated key:", key)

    while True:
        choice = input("[0]: Quit\n[1]: Encrypt\n[2]: Decrypt\nPress 0, 1, or 2: ")
        if choice == '0':
            print("Exiting...")
            sys.exit(0)
        elif choice == '1':
            plaintext = input("Enter the plaintext: ")
            encrypted_text = encrypt(plaintext)
            print(f"Ciphertext: {encrypted_text}")
        elif choice == '2':
            while True:
                cipher = input("Enter a ciphertext (binary format): ")
                if re.match("^(01)+$", cipher):
                    decrypted_text = decrypt(cipher)
                    print(f"Decrypted text: {decrypted_text}")

```

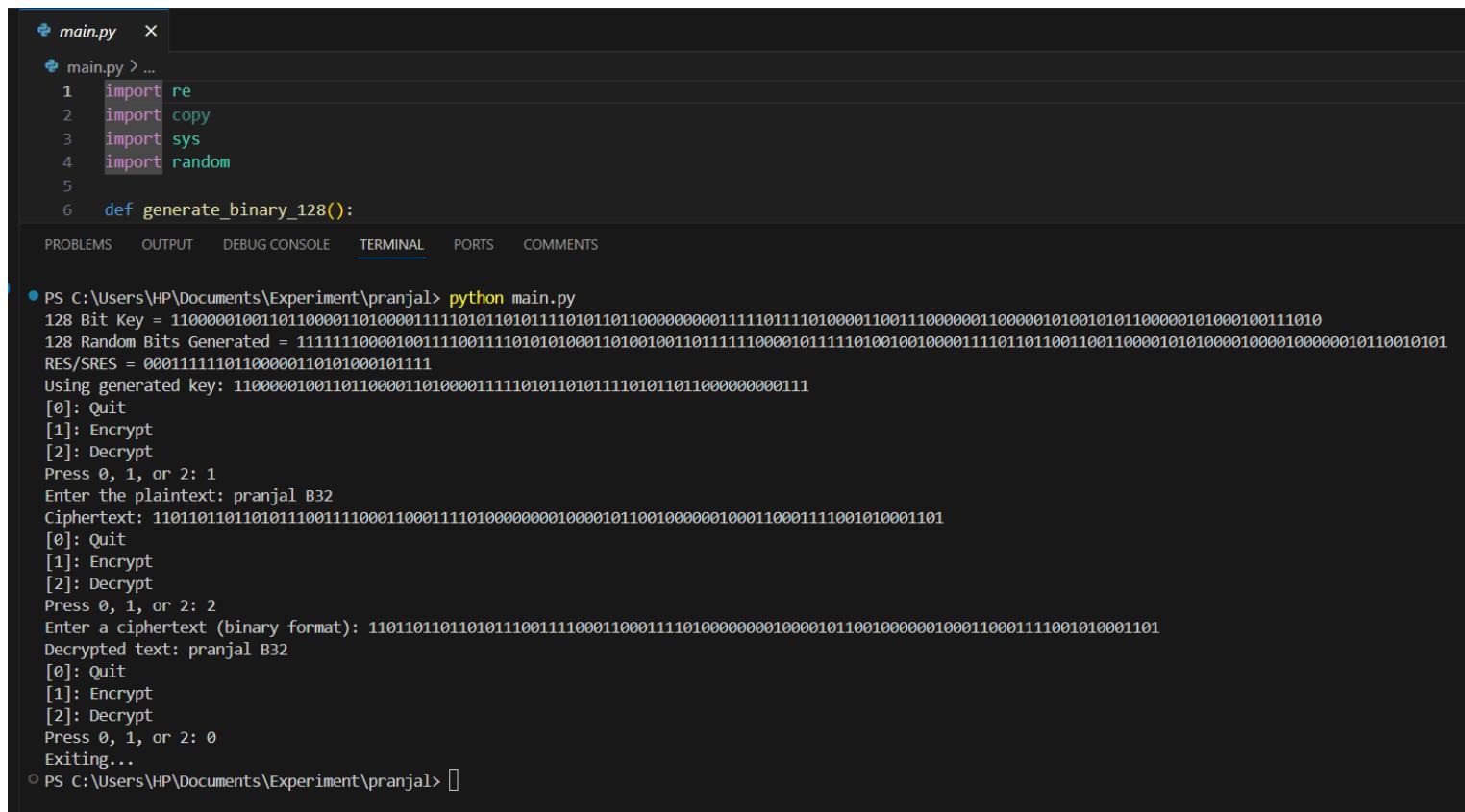
```

        break
else:
    print("Invalid ciphertext. Must be in binary format.")

if __name__ == "__main__":
    main_fun()

```

Q.2: Output of GSM Security Algorithm



```

main.py
main.py > ...
1 import re
2 import copy
3 import sys
4 import random
5
6 def generate_binary_128():

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

● PS C:\Users\HP\Documents\Experiment\pranjal> python main.py
128 Bit Key = 1100001001101000011010000111101011010111010110101100000000011110111101000011001110000011000001010010110000101000100111010
128 Random Bits Generated = 111111100001001110011101010001101001001101111100001011111010010001111011011001100010101000100000010110010101
RES/SRES = 00011111101100000110101000101111
Using generated key: 110000100110100001101000011110101101011110101101100000000111
[0]: Quit
[1]: Encrypt
[2]: Decrypt
Press 0, 1, or 2: 1
Enter the plaintext: pranjal B32
Ciphertext: 110110110110101110011110001100011110100000001000010110010000001000110001111001010001101
[0]: Quit
[1]: Encrypt
[2]: Decrypt
Press 0, 1, or 2: 2
Enter a ciphertext (binary format): 110110110110101110011110001100011110100000001000010110010000001000110001111001010001101
Decrypted text: pranjal B32
[0]: Quit
[1]: Encrypt
[2]: Decrypt
Press 0, 1, or 2: 0
Exiting...
○ PS C:\Users\HP\Documents\Experiment\pranjal>

```

Q.3: List out various elements of GSM architecture and explain in brief function of each element.

The **Global System for Mobile Communications (GSM)** architecture consists of several key components categorized into different subsystems:

1. Mobile Station (MS)

- **Components:** Mobile handset (phone) and SIM (Subscriber Identity Module).
- **Function:**
 - The mobile device allows users to make and receive calls, send SMS, and use data services.

- The SIM stores subscriber-related information, including IMSI (International Mobile Subscriber Identity), authentication key, and network-specific data.

2. Base Station Subsystem (BSS)

- **Components:** Base Transceiver Station (BTS) and Base Station Controller (BSC).
- **Functions:**
 - **BTS:**
 - Handles radio communication with mobile stations.
 - Manages frequency allocation and radio signal processing.
 - **BSC:**
 - Controls multiple BTSs.
 - Manages handovers, power control, and frequency hopping.

3. Network and Switching Subsystem (NSS)

- **Components:**
 - **Mobile Switching Center (MSC)**
 - **Home Location Register (HLR)**
 - **Visitor Location Register (VLR)**
 - **Authentication Center (AuC)**
 - **Equipment Identity Register (EIR)**
- **Functions:**
 - **MSC:**
 - Central switch for call setup, routing, and handovers.
 - Connects GSM network to the Public Switched Telephone Network (PSTN).
 - **HLR:**
 - Stores permanent subscriber information (IMSI, service plans, location, etc.).
 - **VLR:**
 - Temporary database storing subscriber information while in a specific location.
 - **AuC:**
 - Provides security by authenticating users and generating encryption keys.
 - **EIR:**

- Stores information about mobile devices to prevent unauthorized access (blacklist/whitelist database).

4. Operation and Support Subsystem (OSS)

- Components:
 - Operation and Maintenance Center (OMC)
- Function:
 - Manages network operations, including monitoring performance, fault detection, and configuration management.

B.2 Conclusion:

This experiment demonstrates the implementation of a **stream cipher encryption technique** using a **keystream generator** based on a **linear feedback shift register (LFSR)** approach. The process involves:

1. **Random Key Generation:** A 128-bit key is randomly generated and divided into two 64-bit halves, with bitwise XOR operations performed to derive intermediate values for encryption.
2. **Register Loading and Keystream Generation:** Three shift registers (X, Y, Z) are initialized and updated dynamically based on the majority function to produce a keystream.
3. **Encryption & Decryption:**
 1. The plaintext is converted into binary and XORed with the generated keystream to produce ciphertext.
 2. The decryption process reverses the encryption by XORing the ciphertext with the same keystream, successfully retrieving the original message.

Experiment No 4: Set up Wireless Access Point

PART A

(PART A: TO BE REFERRED BY STUDENTS)

A.1 Aim: Set up and configuration of Wireless Access Point.

A.2 Objectives: After successful completion of this experiment students will be able to Set up and configure access point and use it to access internet.

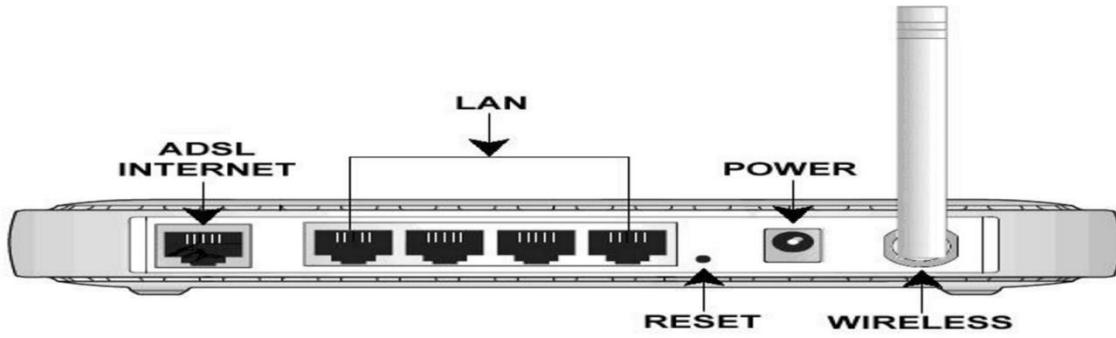
A.3 Outcomes: Student will be able to implement security algorithms for mobile communication network.

A.4 Tools Used/programming language: Cisco Packet Tracer

A.5 Theory:

Access point is a device used to connect Wireless LAN with internet. Access point is used in wireless LAN which might be having standard either IEEE standard or HIPERLAN. It will also be called as router since it works in network layer. Wireless LAN means each computer/laptop connected with each other wirelessly. There are two kinds of LAN possible I) adhoc wireless and ii) infrastructure based Access point configuration is done in such way that it is able to transfer and receive all packets of LAN securely if firewall is installed in that AP.

Access point is will be used here is wifi (D LINK) shown in diagram1 In diagram it is shown that DSL pin where DSL connection will be done where as other port will be used to connect any desktop computer. Following diagram2 will give idea about how access point is seen in wireless LAN Diagram 1



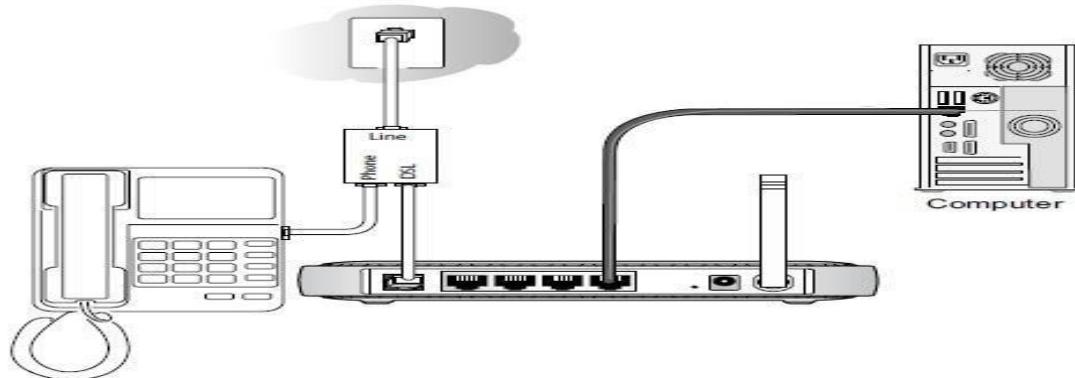
Note: Configuration of AP requires physical connection with internet via ADSL port then start WIFI then Update parameters of AP as per user requirement such as password /LAN name by visiting WIFI router page via internet explorer.

- **Wireless Access Point (WAP):** A **wireless access point (WAP)** is a networking hardware device that allows wireless devices to connect to a wired network using Wi-Fi, or related standards. The AP usually connects to a router (via a wired network) as a standalone device, but it can also be an integral component of the router itself. An AP is differentiated from a hotspot, which is the physical space where the wireless service is provided.



- **Steps for configuration of WAP:**

1. Connect the DSL port of the NETGEAR modem router to the phone line, via the DSL Micro filter, as shown in the diagram. Use an Ethernet cable to connect the computer to any of the four LAN ports as shown in the diagram. Connect the NETGEAR DSL modem router to its power supply unit (PSU) (Not shown in the diagram) and wait about a minute for it to boot up.



2. Open an Internet Explorer browser and type the router IP address which would be either **http://192.168.0.1** or **http://192.168.1.1** in the address bar and press Enter.
 - The Login window will prompt for the router configuration username and password.
 - The default username is **admin** and the default password is **password**
 - The username and password are case sensitive.

- If the default username and password is not working, you might have changed the password. Please try other passwords that you might have changed to. Otherwise, a factory reset is needed to restore the router to factory defaults. To perform a factory reset, see Restoring a NETGEAR home router to the factory default settings.



3. Click **Setup Wizard** on the top left corner, Select **Yes** for the Setup Wizard to detect the type of Internet connection and click **Next**.



4. The Setup Wizard will report which connection type it has discovered, and then display the appropriate configuration page. Please follow the steps under the connection type detected by your router:

Note: If the Setup Wizard finds no connection, please check the physical connection of your devices, and make sure that your ISP has already activated your DSL account.

Wizard Detected PPPoE Login Account Setup

The screenshot shows a window titled "PPPoE". It contains two input fields: "Login" and "Password", each with a corresponding empty text box. Below the fields are three buttons: "Apply", "Cancel", and "Test".

Enter the PPPoE login user name and password. These fields are case sensitive. This information should have been provided to you by your ISP.

Wizard Detected Dynamic IP Account Setup

The screenshot shows a window titled "Dynamic IP Address". It displays a message: "No input data is required." followed by "Click "Apply" to accept this connection method.". Below the message are three buttons: "Apply", "Cancel", and "Test".

Click **Apply** to set Dynamic IP as the connection method.

Wizard Detected IP over ATM Account Setup

IP Over ATM

Internet IP Address

IP Address	[0] . [0] . [0] . [0]
IP Subnet Mask	[0] . [0] . [0] . [0]

Domain Name Server (DNS) Address

Primary DNS	[] . [] . [] . []
Secondary DNS	[] . [] . [] . []

Apply | **Cancel** | **Test**

5.

1. Enter your assigned IP Address and Subnet Mask. This information should have been provided to you by your ISP.
2. Enter the IP address of your ISP's Primary DNS Server. If a Secondary DNS Server address is available, enter it also.
3. Click **Apply** to save the settings.
4. Click **Test** to test your internet connection.

6. Wizard Detected Fixed IP (Static) Account Setup

Fixed IP

Account Name (If Required) []

Domain Name (If Required) []

Internet IP Address

Use Static IP Address

IP Address	[0] . [0] . [0] . [0]
IP Subnet Mask	[0] . [0] . [0] . [0]
Gateway IP Address	[0] . [0] . [0] . [0]

Use IP Over ATM (IPOA)

IP Address	[0] . [0] . [0] . [0]
IP Subnet Mask	[0] . [0] . [0] . [0]
Gateway IP Address	[0] . [0] . [0] . [0]

Domain Name Server (DNS) Address

Primary DNS	[] . [] . [] . []
Secondary DNS	[] . [] . [] . []

Apply | **Cancel** | **Test**

1. If required, enter the **Account Name** and **Domain Name** from your ISP.
2. Choose **Use Static IP Address** or **Use IP Over ATM** (IPoA-RFC1483 Routed) according to the information from your ISP.

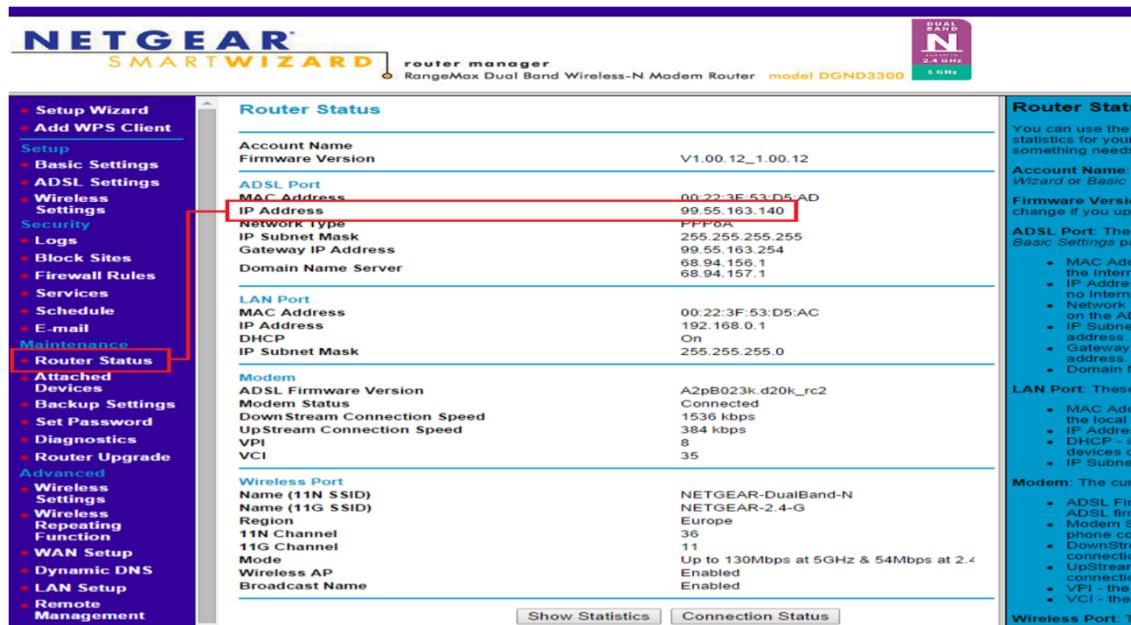
If you choose IPoA, the router will be able to detect the gateway IP address but you still need to provide the router IP address.

3. Enter your assigned **IP Address**, **Subnet Mask**, and the **IP Address** of your ISP's gateway router.

This information should have been provided to you by your ISP.

4. Enter the IP address of your ISP's Primary DNS Server. If a Secondary DNS Server address is available, enter it also. **5. Click Apply** to save the settings.
6. Click **Test** to test your internet connection.

7. The router will now save these settings. When complete, you can verify whether you are connected to the internet from the **Router Status** under **Maintenance** menu.



Verify that you have a valid IP address (not blank or 0.0.0.0) on the Internet or ADSL Port.

8. Procedure on Cisco tracer: Download Cisco tracer from
<https://www.netacad.com/courses/packet-tracer>.
9. Watch you tube video for WI router configuration in Cisco tracer:
<https://www.youtube.com/watch?v=vu0najkh9vQ>

PART B

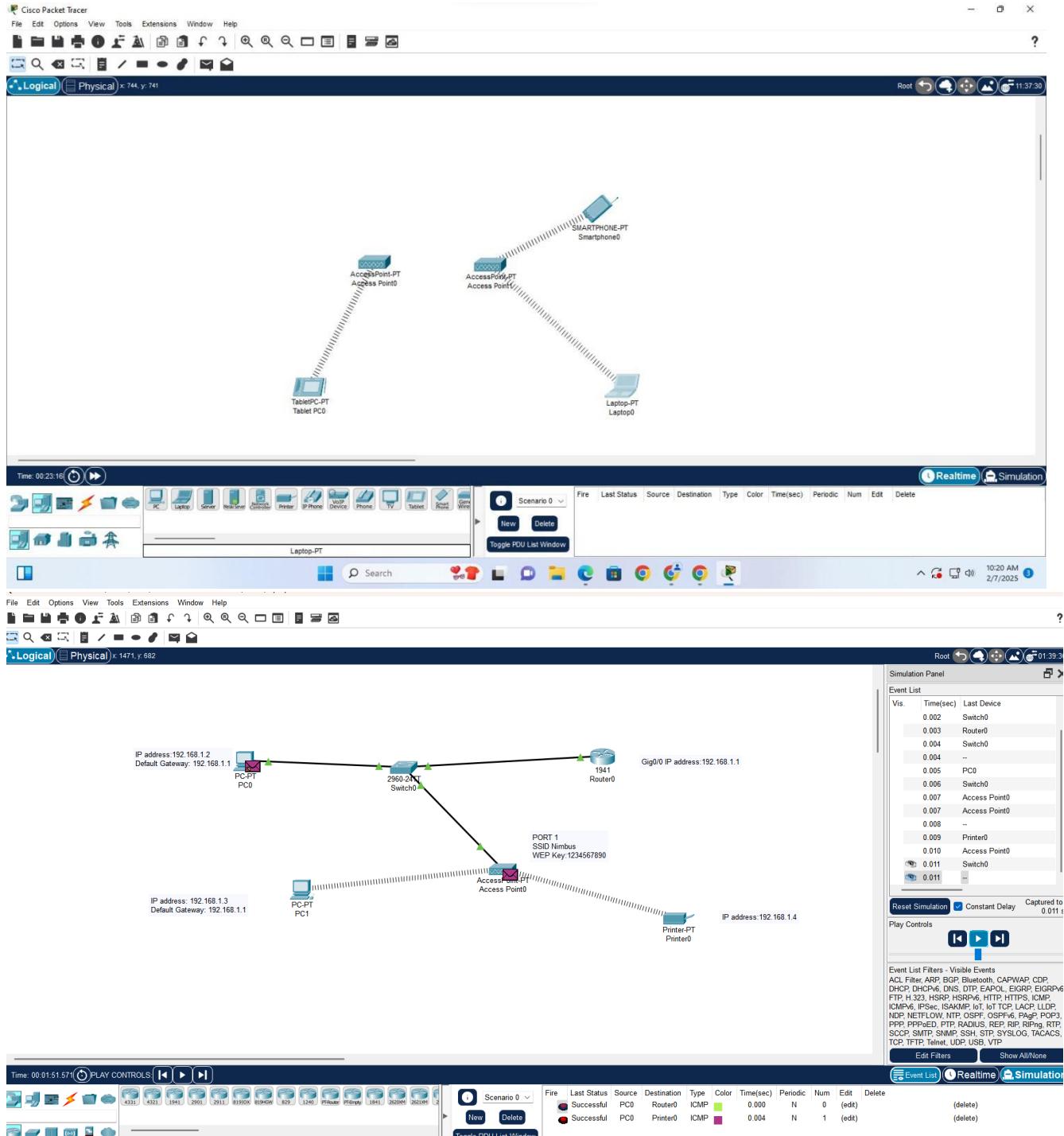
(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the ERP or emailed to the concerned lab in charge faculties at the end of the practical in case there is no ERP access available)

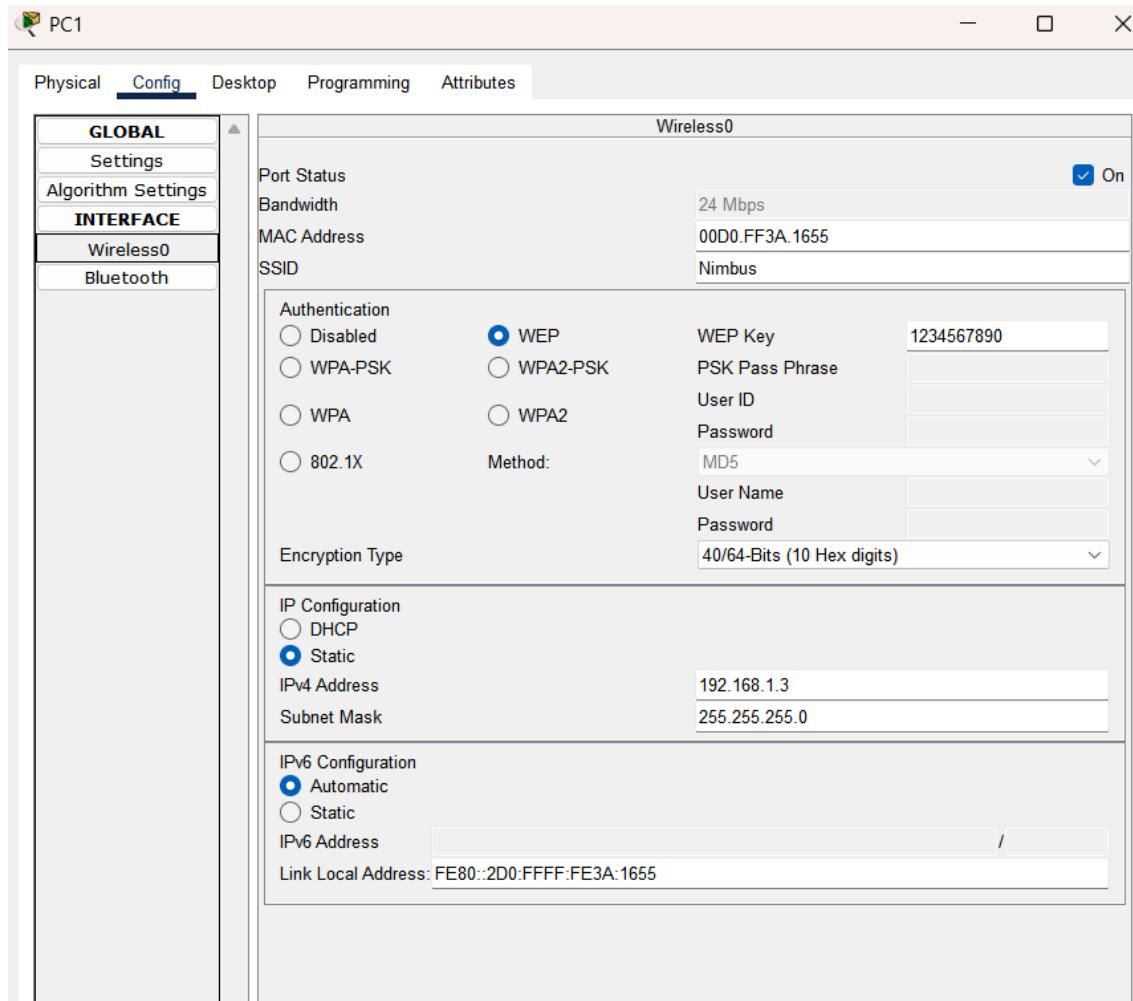
Roll No. B30	Name: Pranjal Bhatt
Class : TE COMPS B	Batch : B2
Date of Experiment:	Date of Submission
Grade :	

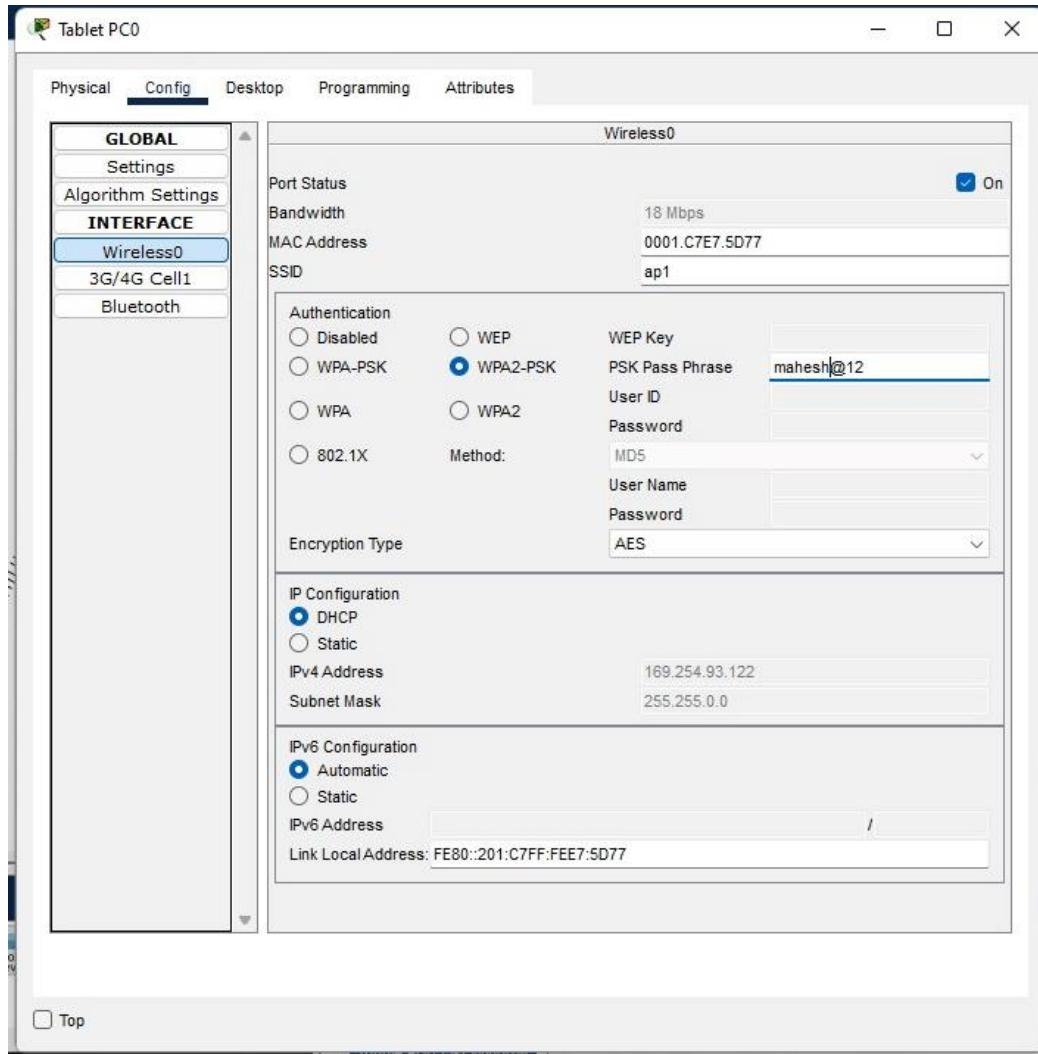
B.1 Question of Curiosity:

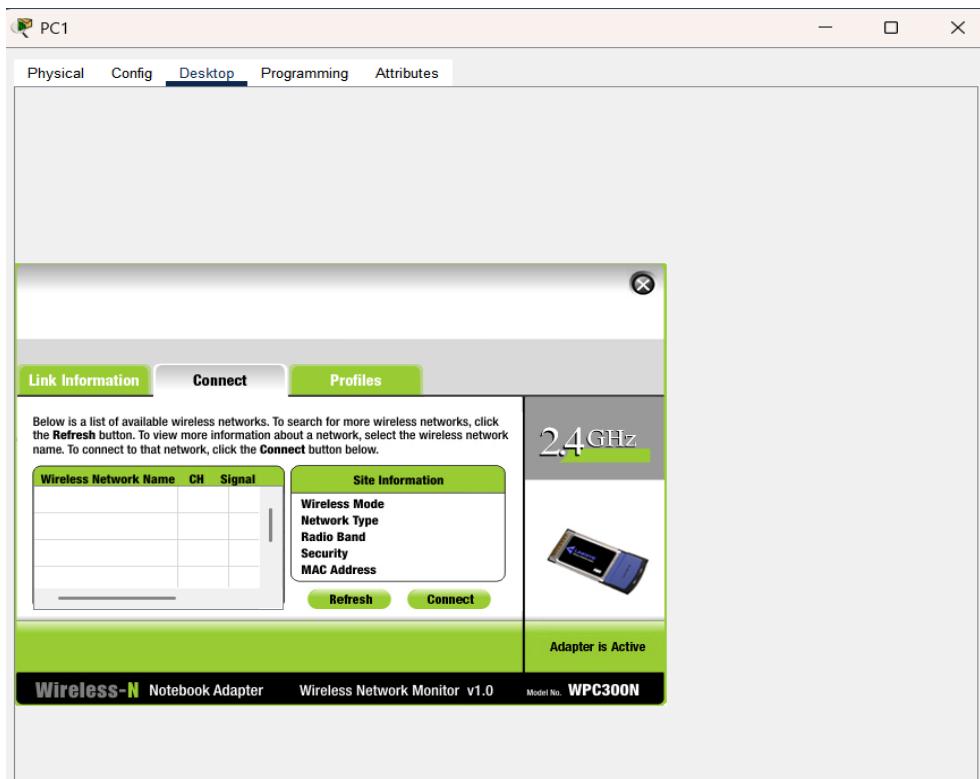
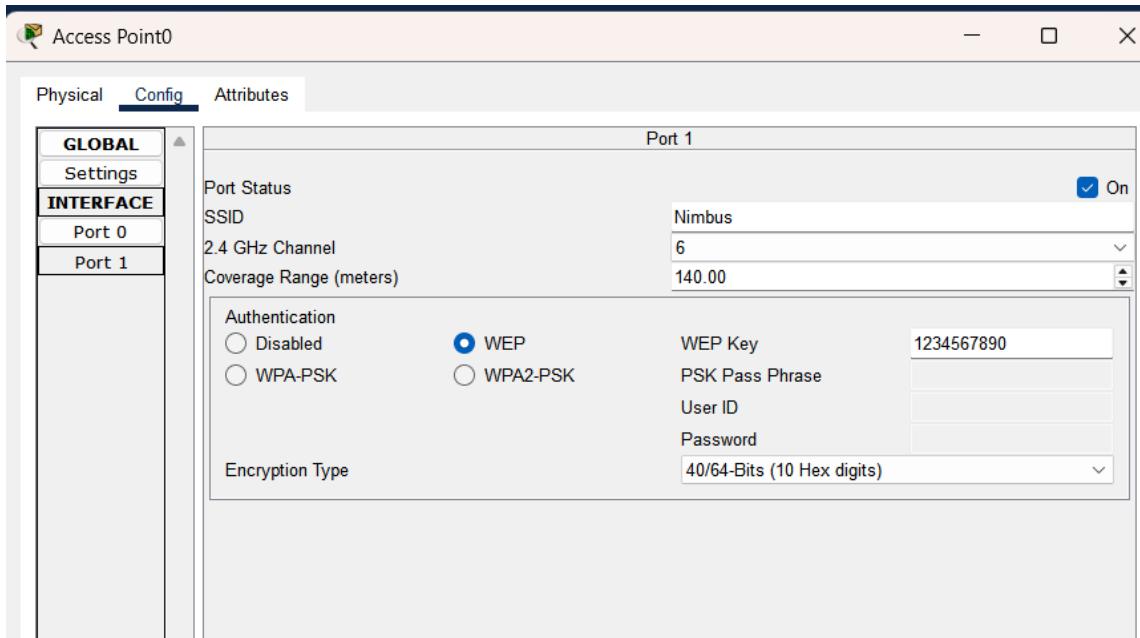
Q.1: Create small Wires less network using WAP. (attach all screenshots)

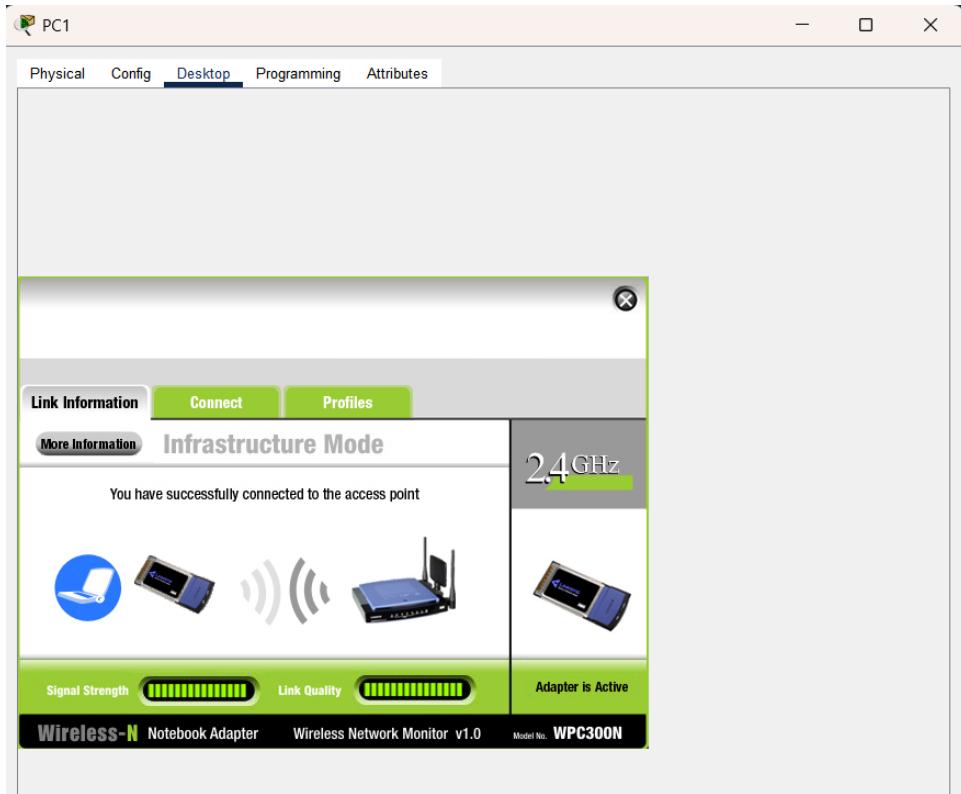


Q.2: How to Perform configuration of wireless access point. (Attach all screenshots)









Q.3: How adhoc Wireless LAN will be created?

An **Ad-Hoc Wireless LAN** (Wireless Local Area Network) is created without requiring a central access point (AP) or router. Devices communicate directly with each other in a peer-to-peer fashion. The steps to create an ad-hoc WLAN are:

1. **Enable Wireless Network Interface:** Ensure that all participating devices have Wi-Fi enabled.
2. **Configure Ad-Hoc Mode:**
 1. In Windows/Linux/macOS, configure the Wi-Fi settings to **Ad-Hoc mode** instead of Infrastructure mode.
 2. Assign a **unique SSID** (network name) for the group.
3. **Set IP Addresses:**
 1. Devices must be assigned static IP addresses in the same subnet (if DHCP is unavailable).
4. **Establish Connections:**
 1. Each device scans for available Ad-Hoc networks and connects to the appropriate one.
5. **Enable File Sharing & Communication:**
 1. Devices can now communicate directly, share files, and form a temporary wireless network.

Q.4: What is difference between WLAN and WiMax ?

Feature	WLAN (Wireless Local Area Network)	WiMax (Worldwide Interoperability for Microwave Access)
Coverage Area	Typically up to 100 meters (indoor/outdoor)	Covers several kilometers (up to 50 km in rural areas)
Technology Used	Uses Wi-Fi (IEEE 802.11) standards	Uses IEEE 802.16 standard
Network Type	Designed for local area networking (home, office, campus)	Designed for wide area networking (broadband internet, rural access)
Speed	Speeds up to 9.6 Gbps (latest Wi-Fi 6E)	Speeds up to 1 Gbps for fixed connections
Infrastructure	Requires an Access Point (AP) or router	Uses Base Stations (BS) similar to cellular towers
Mobility Support	Supports limited mobility within coverage	Supports high mobility similar to cellular networks
Use Case	Home, office, campus networking	Rural broadband, metropolitan area networks, mobile broadband

B.2 Conclusion:

In this experiment, we successfully set up and configured a **Wireless Access Point (WAP)** using **Cisco Packet Tracer**. The experiment demonstrated the step-by-step process of establishing a **wireless LAN (WLAN)**, connecting it to the internet via **DSL**, and configuring essential parameters such as **SSID, security settings, and IP addressing**.

Experiment 05: To implement a Bluetooth network

PART A

A.1 Aim: To implement a Bluetooth network with application as transfer of a file from one device to another.

A.2 Objectives: To understand the security algorithms in mobile networks

A.3 Outcome: Students will be able to articulate the knowledge of GSM, CDMA & Bluetooth technologies and demonstrate it.(LO-2)

A.4 Theory:

Bluetooth is a wireless technology standard used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in the industrial, scientific and medical radio bands, from 2.402 GHz to 2.480 GHz, and building personal area networks (PANs). It was originally conceived as a wireless alternative to RS-232datacables.

Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 35,000member companies in the areas of telecommunication, computing, networking, and consumer electronics. The IEEE standardized Bluetooth as IEEE802.15.1,but no longer maintains the standard. The Bluetooth SIG oversees development of the specification, manages the qualification program, and protects the trademarks. A manufacturer must meet Bluetooth SIG standards to market it as a Bluetooth device.

Transfer of words between two phones using Bluetooth is done below.

A.5Code: Main_Activity.java:<https://github.cm/vinaynpp/mcc> package com.example.bluetooth_communication;import android.app.Dialog;

```
import android.bluetooth.BluetoothA
dapter;import android.content.Intent;
import android.content.pm.PackageMa
nager;import
```

```
android.content.pm.ResolveInfo;import android.net.Uri;  
import android.os.Environment; import  
androidx.appcompat.app.AppCompatActivity;i  
mport android.os.Bundle;  
import  
        android.view.Menu;impor  
t android.view.MenuItem;i  
mport android.view.View; import  
android.widget.AdapterView;i  
mport android.widget.ArrayAdapter;  
import android.widget.Button;  
import  
        android.widget.EditText;import  
android.widget.ListVie
```

```
import android.widget.TextView;
import android.widget.Toast;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    //-----/Create Objects-----
    Button buttonOpenDialog, buttonUp, send;
    TextView textViewFolder;
    EditText dataPath;
    static final int CUSTOM_DIALOG_ID = 0;
    ListView dialogListView;
    File root, currentFolder;
    private List<String> fileList = new ArrayList<String>();
    private static final int DISCOVER_DURATION = 300;
    private static final int REQUEST_BLUETOOTH = 1;
    BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    // @Override de protected void onCreate(Bundle savedInstanceState)
    //-----/On Create-----

    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        dataPath = (EditText) findViewById(R.id.filePath);
        buttonOpenDialog = (Button) findViewById(R.id.opendialog);
        send = (Button) findViewById(R.id.sendBluetooth);
        buttonOpenDialog.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dataPath.setText("");
                showDialog(CUSTOM_DIALOG_ID);
            }
        });
    };

    root = new File(Environment.getExternalStorageDirectory().getAbsolutePath());
    currentFolder = root;
    send.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String folderPath = dataPath.getText().toString();
            if (!folderPath.isEmpty()) {
                File folder = new File(folderPath);
                if (folder.exists() && folder.isDirectory()) {
                    currentFolder = folder;
                } else {
                    Toast.makeText(getApplicationContext(), "Invalid folder path", Toast.LENGTH_SHORT).show();
                }
            }
        }
    });
}

private void showDialog(int id) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Select Action");
    builder.setItems(new String[]{"Open", "Send", "Cancel"}, new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            switch (which) {
                case 0:
                    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
                    intent.setType("*/*");
                    startActivityForResult(intent, REQUEST_BLUETOOTH);
                    break;
                case 1:
                    Intent intent2 = new Intent("android.intent.action.BLUETOOTH_SEND_FILE");
                    intent2.putExtra("file", currentFolder);
                    send.setIntent(intent2);
                    break;
                case 2:
                    dialog.dismiss();
                    break;
            }
        }
    });
    builder.show();
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_BLUETOOTH) {
        if (resultCode == RESULT_OK) {
            Uri uri = data.getData();
            if (uri != null) {
                String filePath = uri.getPath();
                if (!filePath.isEmpty()) {
                    File file = new File(filePath);
                    if (file.exists() && file.isFile()) {
                        String fileName = file.getName();
                        String fileExtension = fileName.substring(fileName.lastIndexOf(".") + 1);
                        String mimeType = MimeTypeMap.getSingleton().getMimeTypeFromExtension(fileExtension);
                        Intent intent = new Intent("android.intent.action.BLUETOOTH_SEND_FILE");
                        intent.putExtra("file", file);
                        intent.putExtra("mimeType", mimeType);
                        send.setIntent(intent);
                    }
                }
            }
        }
    }
}
```

```

        public void onClick(View v){sen
            dViaBluetooth();
        }

    });

}

@Override
protected Dialog onCreateDialog(int id){Dia
    log dialog = null;
    switch(id) { case CUSTOM_DIALOG_ID:
        dialog = new
        Dialog(MainActivity.this);dialog.setContentView
        (R.layout.dailoglayout);dialog.setTitle("File
        Selector");dialog.setCancelable(true);dialog.set
        CanceledOnTouchOutside(true); textFolder
        = (TextView)
        dialog.findViewById(R.id.folder);buttonUp = (Button)
        dialog.findViewById(R.id.up);buttonUp.setOnClickListener(ne
        wView.OnClickListener());
        @Override
        public void onClick(View v)
        {ListDir(curFolder.getParentFile());
        }
    });

}

dialog(ListView) = (ListView)
dialog.findViewById(R.id.dialoglist);dialog(ListView).setOnItemClickListener(newAdapt
erView.OnItemClickListener(){
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
        position, long id){File selected = new File(fileList.get(position));
        if(selected.isDirectory()){Li
            stDir(selected);
        }else if(selected.isFile()){get
            selectedFile(selected);
        }else {
            dismissDialog(CUSTOM_DIALOG_ID);
        }
    }
}

```

```
        }

    });

        break;
    }

    return dialog;
}

@Override
protected void onPrepareDialog(int id, Dialog dialog) {
    super.onPrepareDialog(id, dialog);
    switch (id) {
        case CUSTOM_DIALOG_ID:
            ListDir(curFolder);
            break;
    }
}
```

```

        }

    public void getselectedFile(File
f){dataPath.setText(f.getAbsolutePath());fileLi
st.clear();dismissDialog(CUSTOM_DIALOG_ID);
}

publicvoidListDir(Filef){if(f.
equals(root)){
    buttonUp.setEnabled(false);
}else { buttonUp.setEnabled(true);
}

    curFolder =
f;textFolder.setText(f.getAbsolutePath());
dataPath.setText(f.getAbsolutePath());Fi
le[]files=f.listFiles();fileList.clear();

    for (File file : files)
{fileList.add(file.getPath());
}

ArrayAdapter<String>directoryList=newArrayAdapter<String>(this,andr
oid.R.layout.simple_list_item_1,fileList); dialog_ListView.setAdapter(directoryList);
}

----- //exittoapplication -----
public void exit(View V)
{btAdatper.disable();
    Toast.makeText(this,"***NowBluetoothisoff...Thanks.
***",Toast.LENGTH_LONG).show();finish();}

//Method for send file via bluetoothpublicvoid

sendViaBluetooth(){
if(!dataPath.equals(null)){i
f(btAdatper ==null) {
    Toast.makeText(this,"Devicenotsupport bluetooth",Toast.LENGTH_LONG).show();
}else { enableBluetooth();
}
}
}

```

```
}
```

```
}else{
```

```
        Toast.makeText(this,"Please select a file.",Toast.LENGTH_LONG).show();
    }

}

public void enableBluetooth() {
    Intent discoveryIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
    discoveryIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION,
                           BluetoothAdapter.DISCOVER_DURATION);
    startActivityForResult(discoveryIntent, REQUEST_BLU);
}

// Override method for sending data via bluetooth availability -----
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode == DISCOVER_DURATION && requestCode == REQUEST_BLU) {
        Intent intent = new Intent();
        intent.setAction(Intent.ACTION_SEND);
        intent.setType("*/*");
        File file = new File(dataPath.getText().toString());
        intent.putExtra(Intent.EXTRA_STREAM, Uri.fromFile(file));
        PackageManager pm = getPackageManager();
        List<ResolveInfo> list = pm.queryIntentActivities(intent, 0);
        if (list.size() > 0) {
            String packageName = null;
            String className = null;
            boolean found = false;
            for (ResolveInfo info : list) {
                packageName = info.activityInfo.packageName;
```

```
if(packageName.equals("com.android.bluetooth")){cl  
    a ssName=info.activityInfo.name; found=true;b  
    reak;  
}  
  
}  
  
//CHECKBLUETOOTHavailableornot----- if(!found){  
    Toast.makeText(this,"Bluetoothnotbeenfound",Toast.LENGTH_LONG).show();  
}else {  
    i. setClassName(packageName,className);  
    startActivity(i);
```

```
}

}

        }else {
    Toast.makeText(this,"Bluetooth is cancelled",Toast.LENGTH_LONG).show();
}

}

@Override
public boolean onCreateOptionsMenu(Menu menu){
    // Inflate the menu; this adds items to the action bar if it is
    getMenuInflater().inflate(R.menu.menu_main,menu); return true;
}

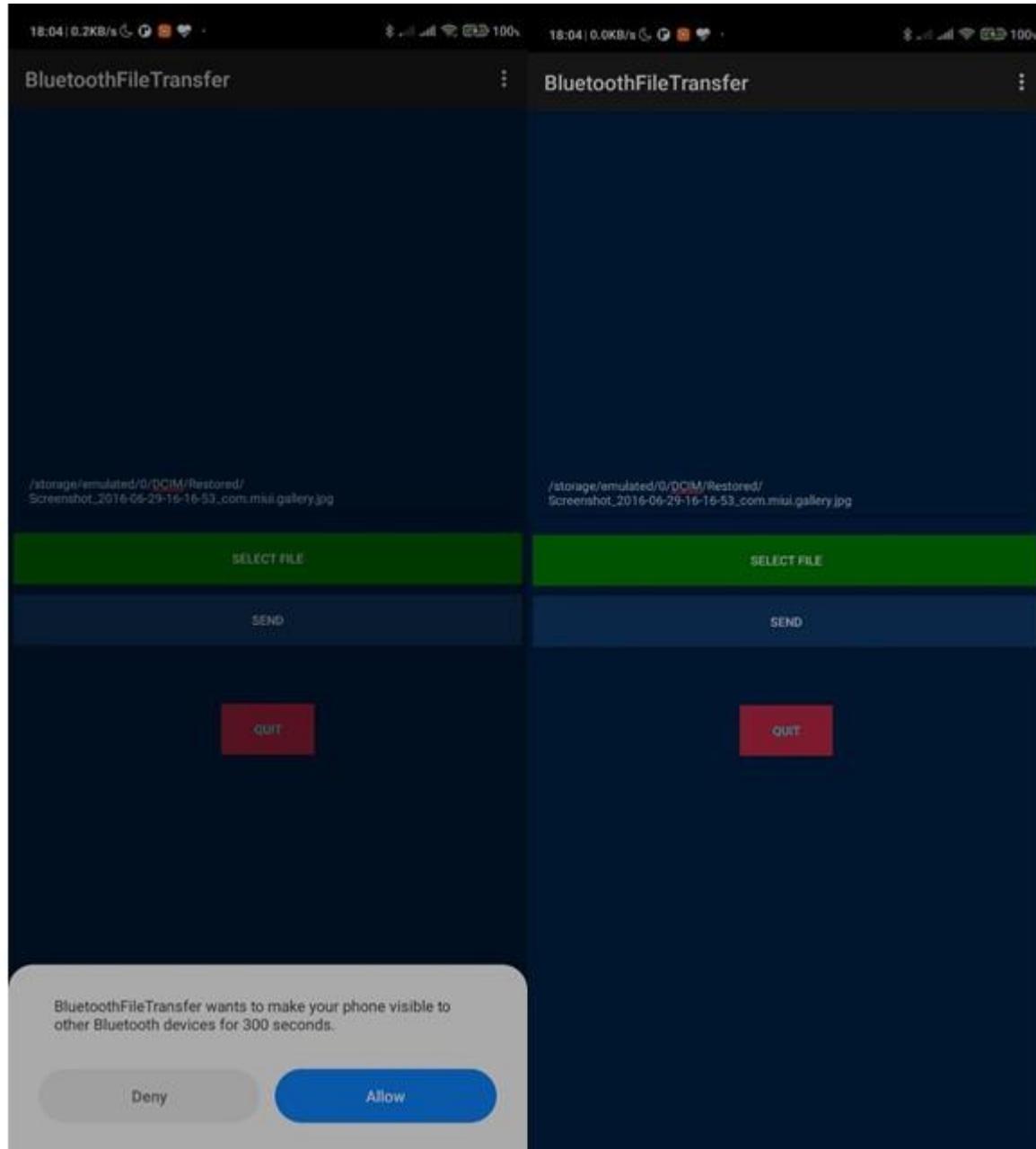
@Override
public boolean onOptionsItemSelected(MenuItem item){
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml. int id = item.getItemId();

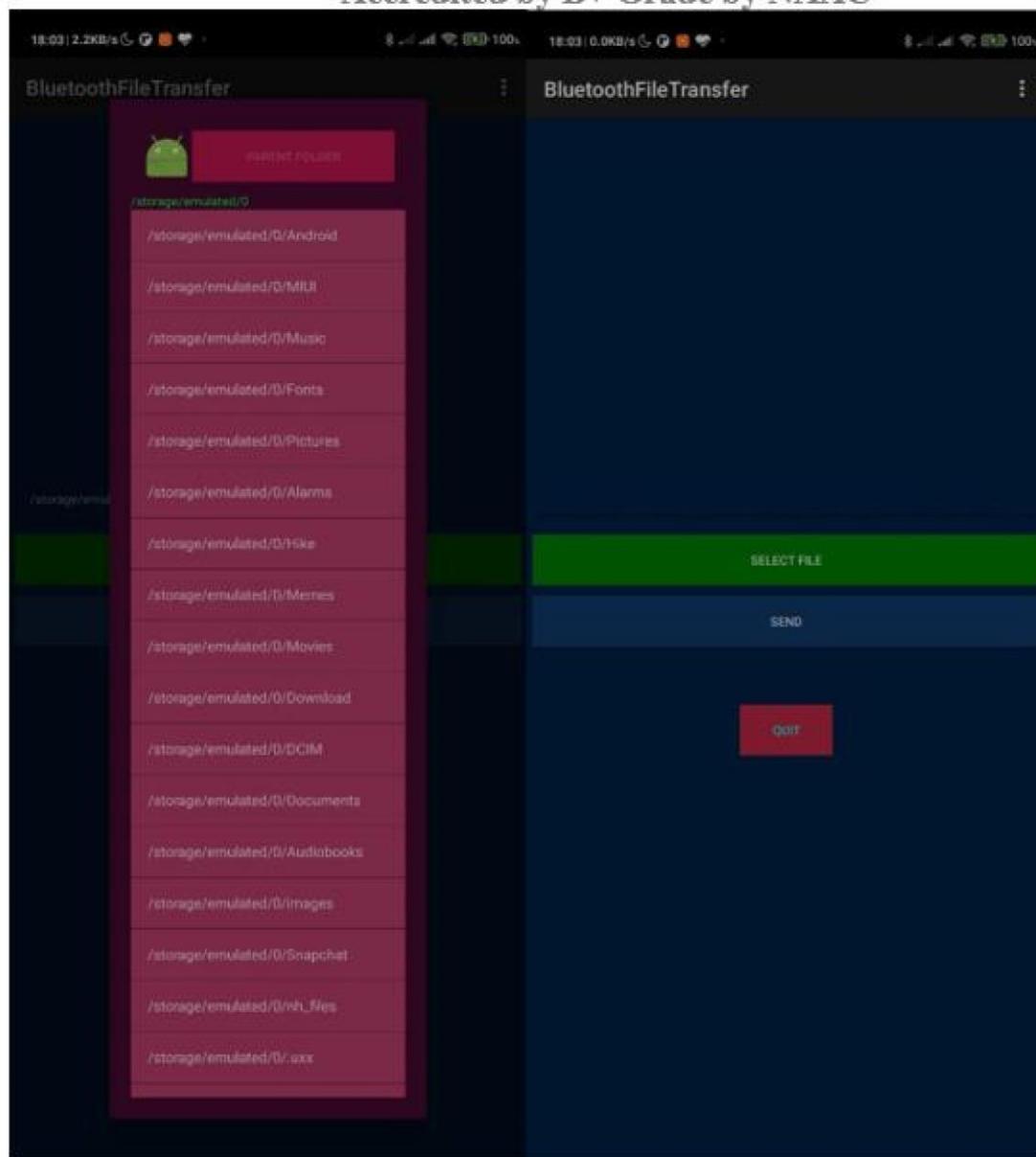
    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        Toast.makeText(this, "*****\nDeveloper: Santosh Kumar
Singh\nContact: superssingh@gmail.com\n*****",Toast.LENGTH_LONG).show();
        return true;
    }

    return super.onOptionsItemSelected(item);
}

}
```

Output:





PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll. No. B30	Name: Pranjal Bhatt
Class :TE COMPS B	Batch: B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student/steps:

MainActivity.java:

```
package com.example.filetransferbluetooth;

import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.net.Uri;
```

```
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Set;

public class MainActivity extends Activity {

    private static final int REQUEST_ENABLE_BT = 1;
    private static final int REQUEST_SELECT_FILE = 2;

    private BluetoothAdapter bluetoothAdapter;
    private ArrayAdapter<String> deviceListAdapter;
    private String selectedDeviceAddress;
    private Uri selectedFileUri;

    private TextView selectedDeviceText;
    private TextView selectedFileText;
    private Button sendFileBtn;

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

        Button enableBluetoothBtn = findViewById(R.id.enableBluetoothBtn);
        selectedDeviceText = findViewById(R.id.selectedDeviceText);
        ListView devicesListView = findViewById(R.id.devicesListView);
        Button selectFileBtn = findViewById(R.id.selectFileBtn);
        selectedFileText = findViewById(R.id.selectedFileText);
```

```
sendFileBtn = findViewById(R.id.sendFileBtn);

bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
deviceListAdapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1);
devicesListView.setAdapter(deviceListAdapter);

// Enable Bluetooth
enableBluetoothBtn.setOnClickListener(view -> {
    if (bluetoothAdapter == null) {
        Toast.makeText(this, "Bluetooth not supported on this device",
        Toast.LENGTH_SHORT).show();
        return;
    }
    if (!bluetoothAdapter.isEnabled()) {
        Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
    } else {
        Toast.makeText(this, "Bluetooth is already enabled", Toast.LENGTH_SHORT).show();
        listPairedDevices();
    }
});

// List paired Bluetooth devices
devicesListView.setOnItemClickListener((adapterView, view, position, id) -> {
    String deviceInfo = ((TextView) view).getText().toString();
    selectedDeviceAddress = deviceInfo.substring(deviceInfo.length() - 17);
    selectedDeviceText.setText("Selected Device: " + selectedDeviceAddress);
});

// Select a file
selectFileBtn.setOnClickListener(view -> {
    Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
    intent.setType("*/*");
    startActivityForResult(Intent.createChooser(intent, "Select File"), REQUEST_SELECT_FILE);
});

// Send file
```

```
sendFileBtn.setOnClickListener(view -> {
    if (selectedDeviceAddress == null || selectedFileUri == null) {
        Toast.makeText(this, "Please select a device and file first", Toast.LENGTH_SHORT).show();
        return;
    }
    sendData(selectedDeviceAddress, selectedFileUri);
});

// Register Bluetooth discovery receiver
IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
registerReceiver(blueoothReceiver, filter);
}

@SuppressWarnings("MissingPermission")
private void listPairedDevices() {
    @SuppressWarnings("MissingPermission") Set<BluetoothDevice> pairedDevices =
    bluetoothAdapter.getBondedDevices();
    deviceListAdapter.clear();
    if (pairedDevices.size() > 0) {
        for (BluetoothDevice device : pairedDevices) {
            deviceListAdapter.add(device.getName() + "\n" + device.getAddress());
        }
    } else {
        deviceListAdapter.add("No paired devices found");
    }
}

private void sendData(String deviceAddress, Uri fileUri) {
    Intent intent = new Intent();
    intent.setAction(Intent.ACTION_SEND);
    intent.setType("*/*");
    intent.putExtra(Intent.EXTRA_STREAM, fileUri);
    intent.setPackage("com.android.bluetooth");
    startActivity(Intent.createChooser(intent, "Send File"));
}

@Override
```

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == REQUEST_ENABLE_BT && resultCode == RESULT_OK) {
        Toast.makeText(this, "Bluetooth enabled", Toast.LENGTH_SHORT).show();
        listPairedDevices();
    } else if (requestCode == REQUEST_SELECT_FILE && resultCode == RESULT_OK && data != null)
    {
        selectedFileUri = data.getData();
        selectedFileText.setText("Selected File: " + selectedFileUri.getPath());
    }
}

private final BroadcastReceiver bluetoothReceiver = new BroadcastReceiver() {
    @SuppressLint("MissingPermission")
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();
        if (BluetoothDevice.ACTION_FOUND.equals(action)) {
            BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
            if (device != null) {
                deviceListAdapter.add(device.getName() + "\n" + device.getAddress());
            }
        }
    }
};

@Override
protected void onDestroy() {
    super.onDestroy();
    unregisterReceiver(bluetoothReceiver);
}

}

AndroidManifest.xml:
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.filetransferbluetooth">
```

```
<!-- Bluetooth and Storage Permissions -->
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />
<uses-permission android:name="android.permission.BLUETOOTH_SCAN" />
<uses-permission android:name="android.permission.BLUETOOTH_ADVERTISE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

<!-- Bluetooth features -->
<uses-feature android:name="android.hardware.bluetooth" android:required="true"/>
<uses-feature android:name="android.hardware.bluetooth_le" android:required="false"/>

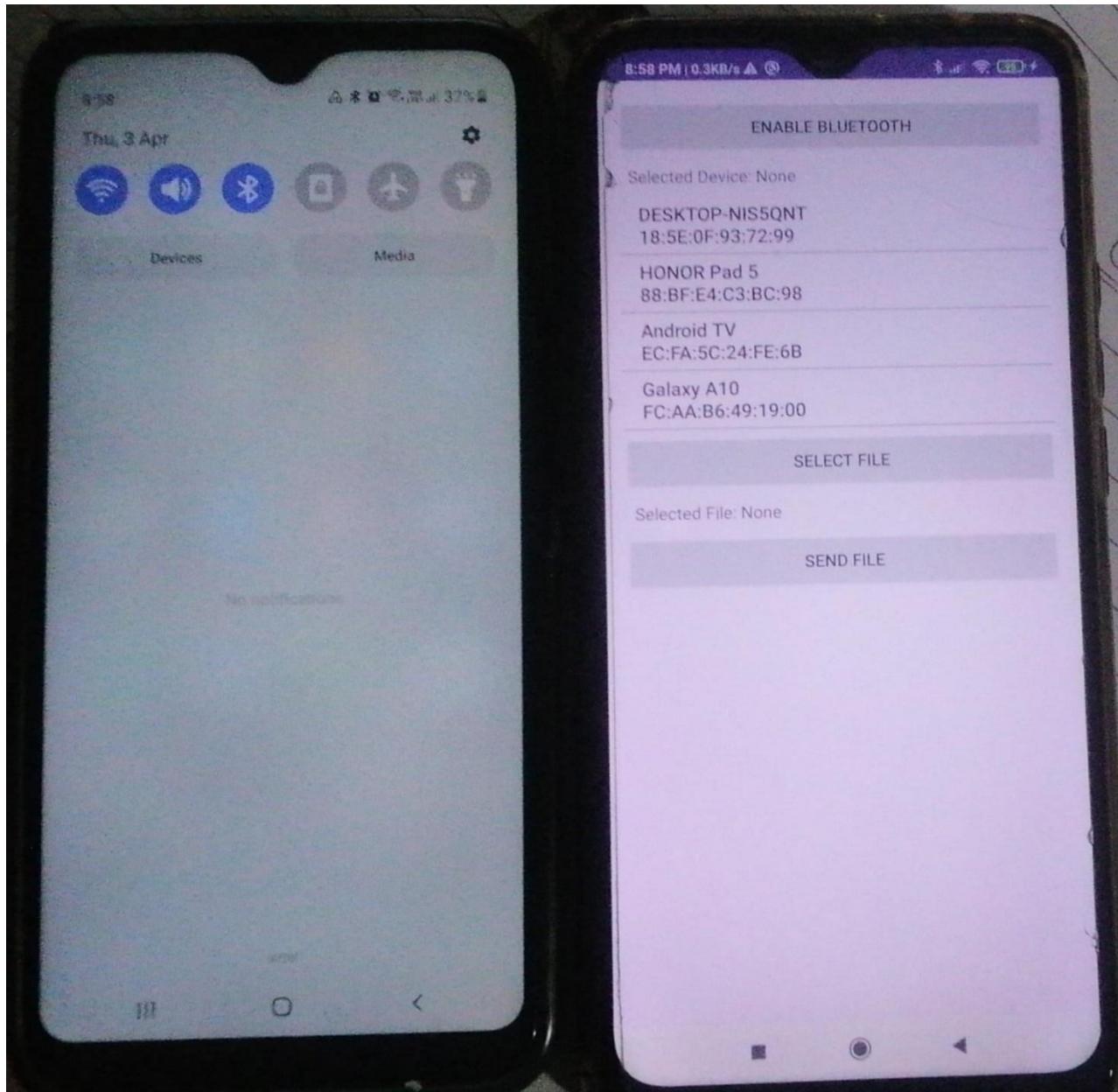
<application
    android:allowBackup="true"
    android:theme="@style/Theme.Filetransferbluetooth">

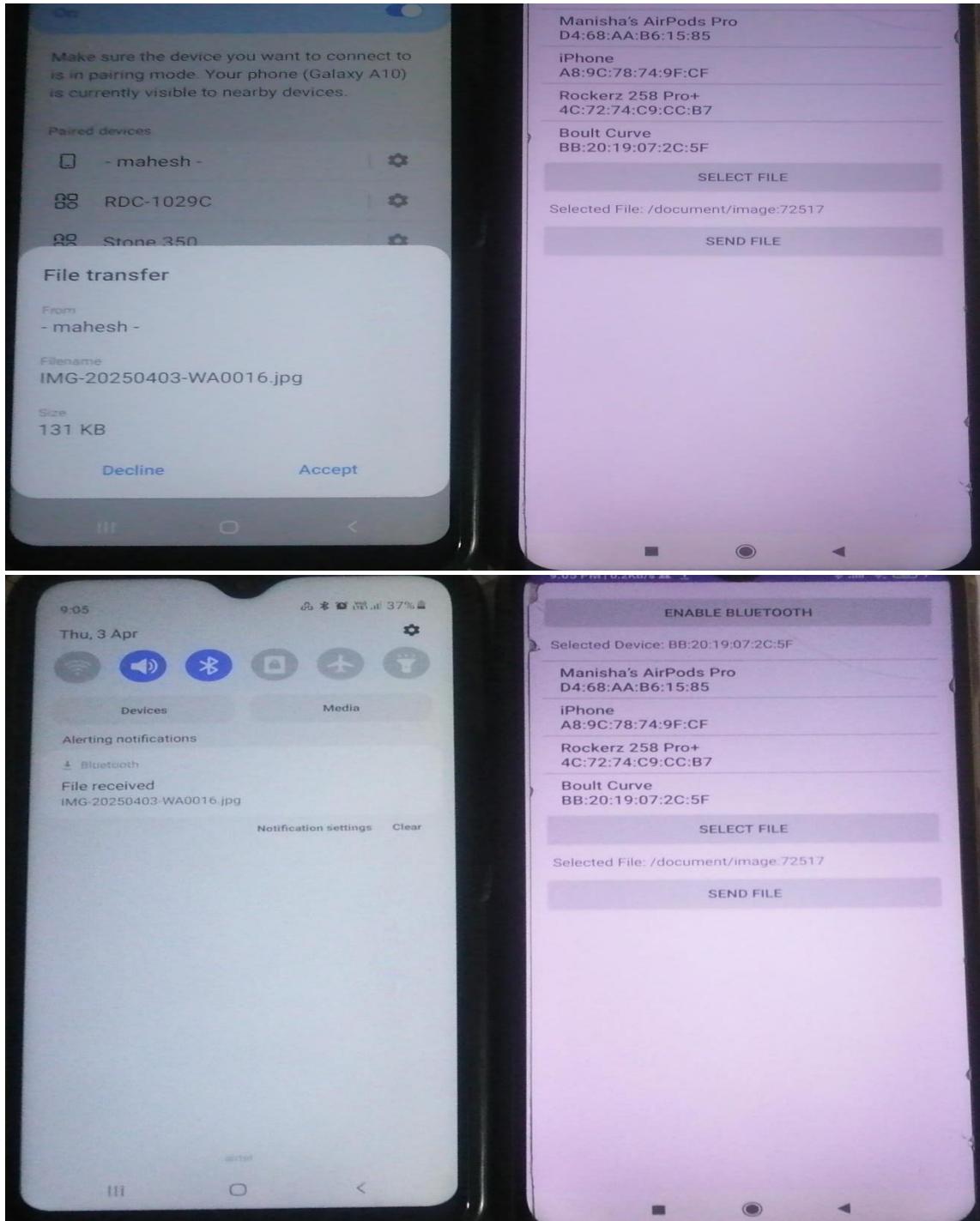
    <activity android:name=".MainActivity"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>

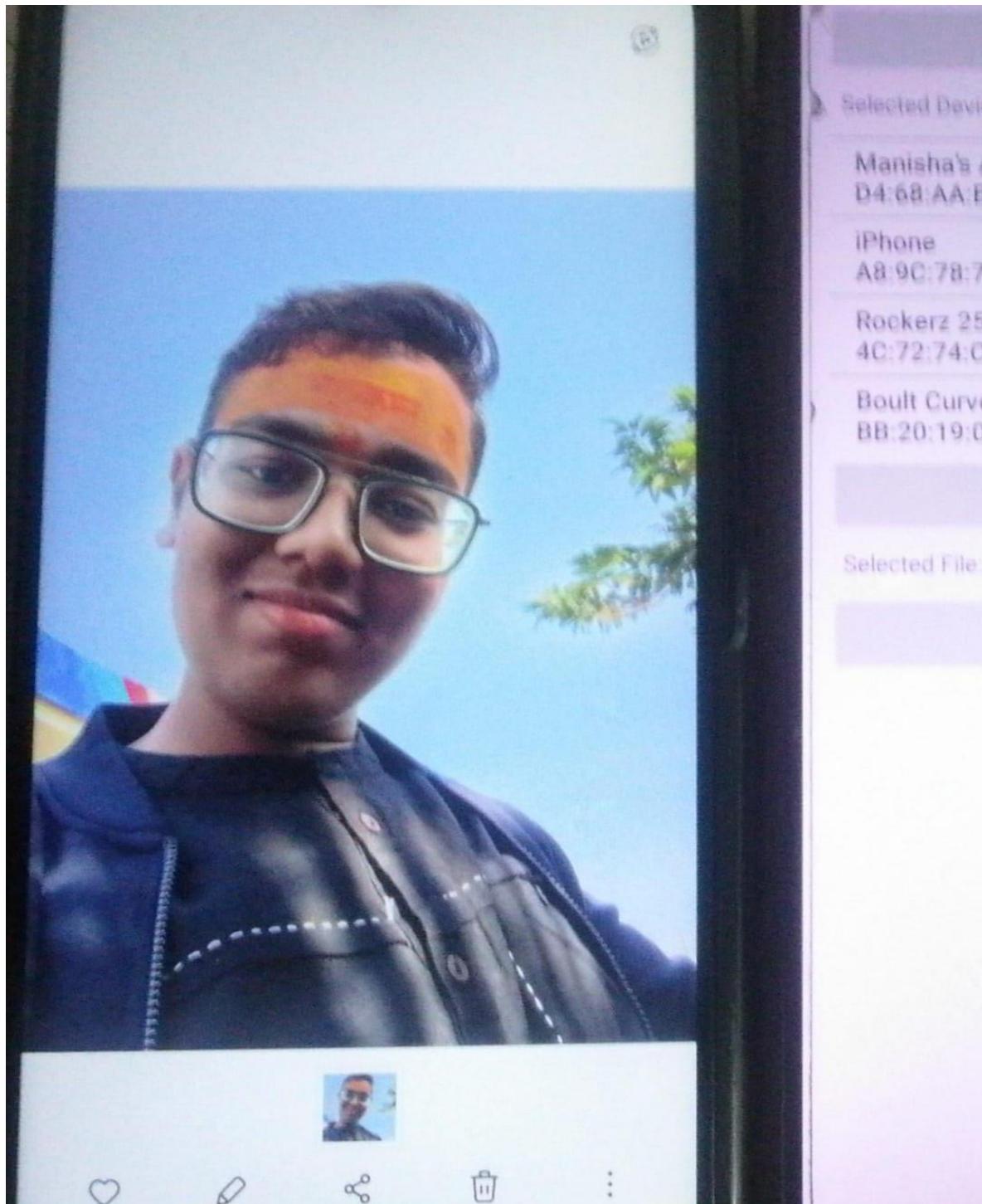
</application>

</manifest>
```

B.2 Input and Output:







B.3 Observations and learning:

During the experiment, a Bluetooth network was implemented to facilitate the transfer of a file between two devices. The connection was established using Bluetooth pairing, and the file was successfully transferred. Various security mechanisms such as authentication and encryption were observed in action, ensuring secure communication between the devices. Additionally, factors such as range, interference, and transfer speed were noted as affecting performance.

B.4 Conclusion:

The experiment demonstrated the practical application of Bluetooth technology in file transfer. It provided insights into Bluetooth security mechanisms and how they function within mobile networks. Understanding Bluetooth architecture, piconet, and scatternet helped in comprehending the broader application of short-range wireless communication in real-world scenarios.

B.5 Question of Curiosity

1) Explain Bluetooth architecture?

Bluetooth architecture consists of the following components:

1. **Radio Layer** – Responsible for physical transmission using frequency hopping spread spectrum (FHSS).
2. **Baseband Layer** – Handles connection establishment, addressing, and packet formats.
3. **Link Manager Protocol (LMP)** – Manages link setup, authentication, encryption, and power control.
4. **Logical Link Control and Adaptation Protocol (L2CAP)** – Provides connection-oriented and connectionless data services to upper layers.
5. **Service Discovery Protocol (SDP)** – Helps devices discover available Bluetooth services.
6. **Host Controller Interface (HCI)** – Acts as an interface between hardware and software, allowing communication between Bluetooth devices and the host system.
7. **Application Layer** – Includes software applications that use Bluetooth for file transfer, audio streaming, or other functionalities.

2) What is piconet and scatternet?

- **Piconet:** A piconet is a small network consisting of one master device and up to seven active slave devices connected via Bluetooth. The master device controls communication, while the slaves follow its instructions. Piconets are dynamically established and can change as devices join or leave.
- **Scatternet:** A scatternet is a network of multiple interconnected piconets, where a device can act as a bridge by participating in two or more piconets simultaneously. This allows greater coverage and more device connectivity.

Experiment 06: To develop an android application that creates an alert upon receiving a message

PART A

A.1 Aim: To develop an android application that creates an alert upon receiving a message.

A.2 Objectives: To introduce students with various tools like Android Studio, NS2, Wireshark, Cisco packet tracer, WAP supported browser etc.

A.3 Outcomes:

After successful completion of this experiment students will be able to develop an android application that creates an alert upon receiving a message

A.4 Theory:

SOFTWARE:

- Android Studio
- The Android SDK (Starter Package)
- Gradle
- Java Development Kit (JDK) 5

DESCRIPTION:

- Open android studio and select new android project . • Give project name and select next
- Choose the android version.
- Enter the package name. package name must be two word separated by comma and click finish
- Go to package explorer in the left hand side and select our project . • Go to res folder and select layout. Double click the main.xml file

- Now you can see the Graphics layout window.

INPUT:

MAINACTIVITY.JAVA

```
package com.alert;

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

public class MainActivity extends AppCompatActivity{ Button notify; EditText
e;

@Override
        protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);
notify= (Button) findViewById(R.id.button); e= (EditText) findViewById(R.id.editText);

notify.setOnClickListener(new View.OnClickListener() { @Override public
void onClick(View v) {
Intent intent = new Intent(MainActivity.this, SecondActivity.class);
PendingIntent pending = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);
Notification noti = new Notification.Builder(MainActivity.this).setContentTitle("New
Message").setContentText(e.getText().toString()).setSmallIcon(R.mipmap.ic_l
auncher).setContentIntent(pending).build();
        NotificationManager manager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
noti.flags |= Notification.FLAG_AUTO_CANCEL; manager.notify(0, noti);
}
}
```

```
});  
}  
}
```

ACTIVITY MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>  
    android:layout_margin="10dp" android:orientation="vertical">  
  
        <TextView  
            android:layout_width="wrap_content"      android:layout_height="wrap_content"  
            android:text="Message" android:textSize="30sp" />  
  
        <EditText  
            android:id="@+id/editText" android:layout_width="match_parent"  
            android:layout_height="wrap_content" android:singleLine="true" android:textSize="30sp" />  
  
        <Button  
            android:id="@+id/button"                  android:layout_width="wrap_content"  
            android:layout_height="wrap_content" android:layout_margin="30dp"  
            android:layout_gravity="center" android:text="Notify" android:textSize="30sp"/> </LinearLayout>
```

ANDROIDMANIFEST.XML

```
<?xml version="1.0" encoding="utf-8"?>  
    <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.alert">  
  
        <application  
            android:allowBackup="true" android:icon="@mipmap/ic_launcher"  
            android:label="@string/app_name" android:roundIcon="@mipmap/ic_launcher_round"  
            android:supportsRtl="true" android:theme="@style/Theme.Alert">  
            <activity  
                android:name=".SecondActivity" android:exported="true" />  
            <activity  
                android:name=".MainActivity" android:exported="true">
```

```
<intent-filter>
<action android:name="android.intent.action.MAIN" />

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

CREATING A NEW ACTIVITY

Create a new activity named “SecondActivity”

File -> New -> Activity -> Empty Activity

SECONDACTIVITY.JAVA

```
package com.alert;

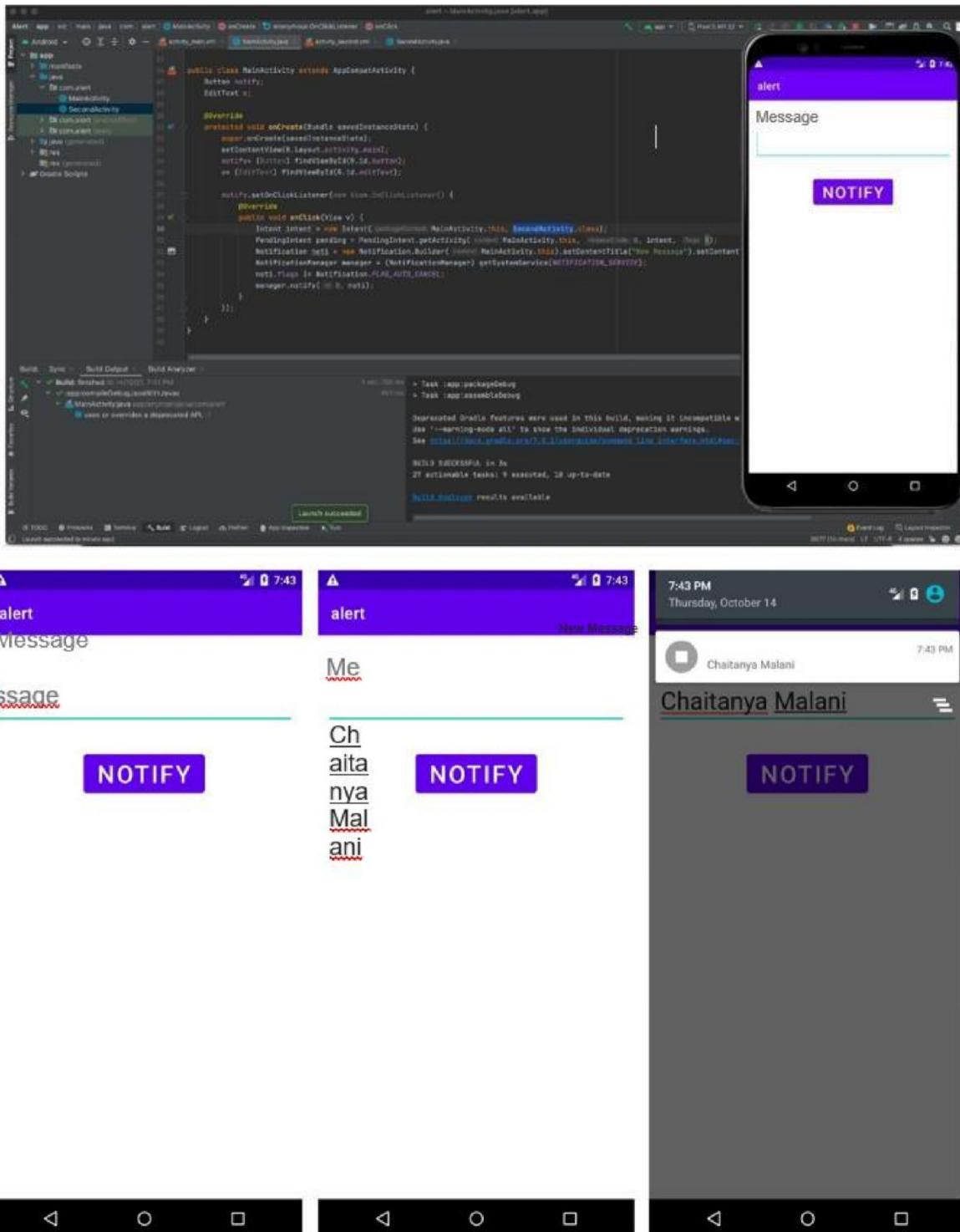
import androidx.appcompat.app.AppCompatActivity; import android.os.Bundle; public
class SecondActivity extends AppCompatActivity{ @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState); setContentView(R.layout.activity_second);
}
}
```

ACTIVITY SECOND.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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=".SecondActivity">

</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:



PART B**(PART B: TO BE COMPLETED BY STUDENTS)**

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll. No:B30	Name: Pranjal Bhatt
Class :TE COMPS B	Batch:B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student/steps:**MainActivity.java:**

```
package com.example.alert;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.app.NotificationCompat;
import androidx.core.app.NotificationManagerCompat;

import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
```

```
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    Button notify;
    EditText e;
    private final String CHANNEL_ID = "alert_channel";

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

        notify = findViewById(R.id.button);
        e = findViewById(R.id.editText);

        // Request notification permission (Android 13+)
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
            if (!ActivityCompat.checkSelfPermission(this,
                android.Manifest.permission.POST_NOTIFICATIONS)
                != PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(this,
                    new String[]{android.Manifest.permission.POST_NOTIFICATIONS}, 1);
            }
        }

        // Create notification channel (needed for Android 8.0+)
        createNotificationChannel();

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

    private void sendNotification() {
        Intent intent = new Intent(this, NotificationService.class);
        startService(intent);
    }

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT < Build.VERSION_CODES.O) {
            return;
        }

        String name = "Alert Channel";
        String description = "A channel for displaying important alerts";
        int importance = NotificationManager.IMPORTANCE_HIGH;
        Uri soundUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);

        NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);
        channel.setDescription(description);
        channel.setSound(soundUri, NotificationManager.RINGER_MODE_NORMAL);

        NotificationManager manager = getSystemService(NotificationManager.class);
        manager.createNotificationChannel(channel);
    }
}
```

```
        }
    });
}

private void createNotificationChannel() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = "Alert Notifications";
        String description = "Includes all alert notifications";
        int importance = NotificationManager.IMPORTANCE_HIGH; // Changed to HIGH

        NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name, importance);
        channel.setDescription(description);

        NotificationManager notificationManager = getSystemService(NotificationManager.class);
        notificationManager.createNotificationChannel(channel);
    }
}

private void sendNotification() {
    String message = e.getText().toString().trim(); // Remove unnecessary spaces

    Intent intent = new Intent(MainActivity.this, SecondActivity.class);
    intent.putExtra("message", message); // Ensure the message is passed

    PendingIntent pendingIntent = PendingIntent.getActivity(
        MainActivity.this,
        0,
        intent,
        PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE
    );

    NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.mipmap.ic_launcher)
        .setContentTitle("New Message")
        .setContentText(message.isEmpty() ? "No message entered" : message) // Avoid null text
        .setPriority(NotificationCompat.PRIORITY_HIGH) // Changed to HIGH
        .setAutoCancel(true)
```

```
.setContentIntent(pendingIntent);

NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
    if (ActivityCompat.checkSelfPermission(this,
            android.Manifest.permission.POST_NOTIFICATIONS) != PackageManager.PERMISSION_GRANTED) {
        return;
    }

    int notificationId = (int) System.currentTimeMillis(); // Unique notification ID
    notificationManager.notify(notificationId, builder.build());
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == 1) {
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            // Permission granted
        } else {
            // Permission denied
        }
    }
}
```

AndroidManifest.xml:

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

    <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
```

```
        android:supportsRtl="true"
        android:theme="@style/Theme.Alert"
        tools:targetApi="31">
    <activity
        android:name=".SecondActivity"
        android:exported="false" />
    <activity
        android:name=".MainActivity"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

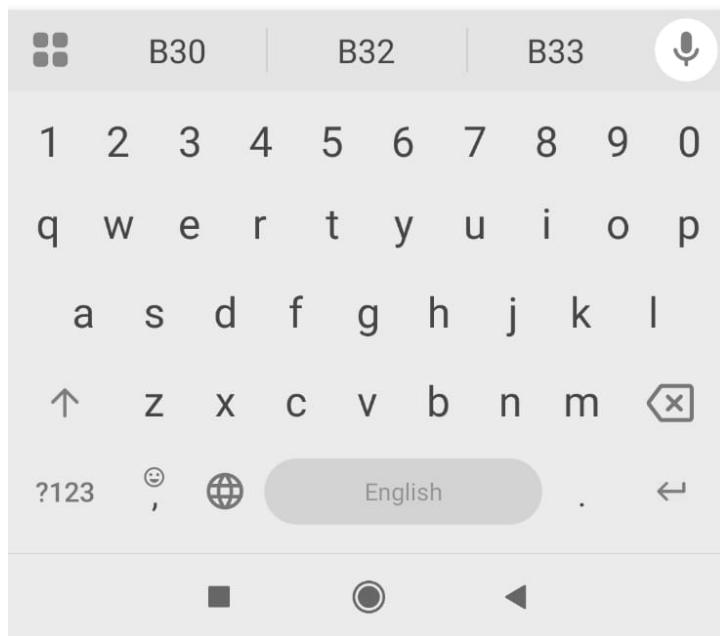
</manifest>
```

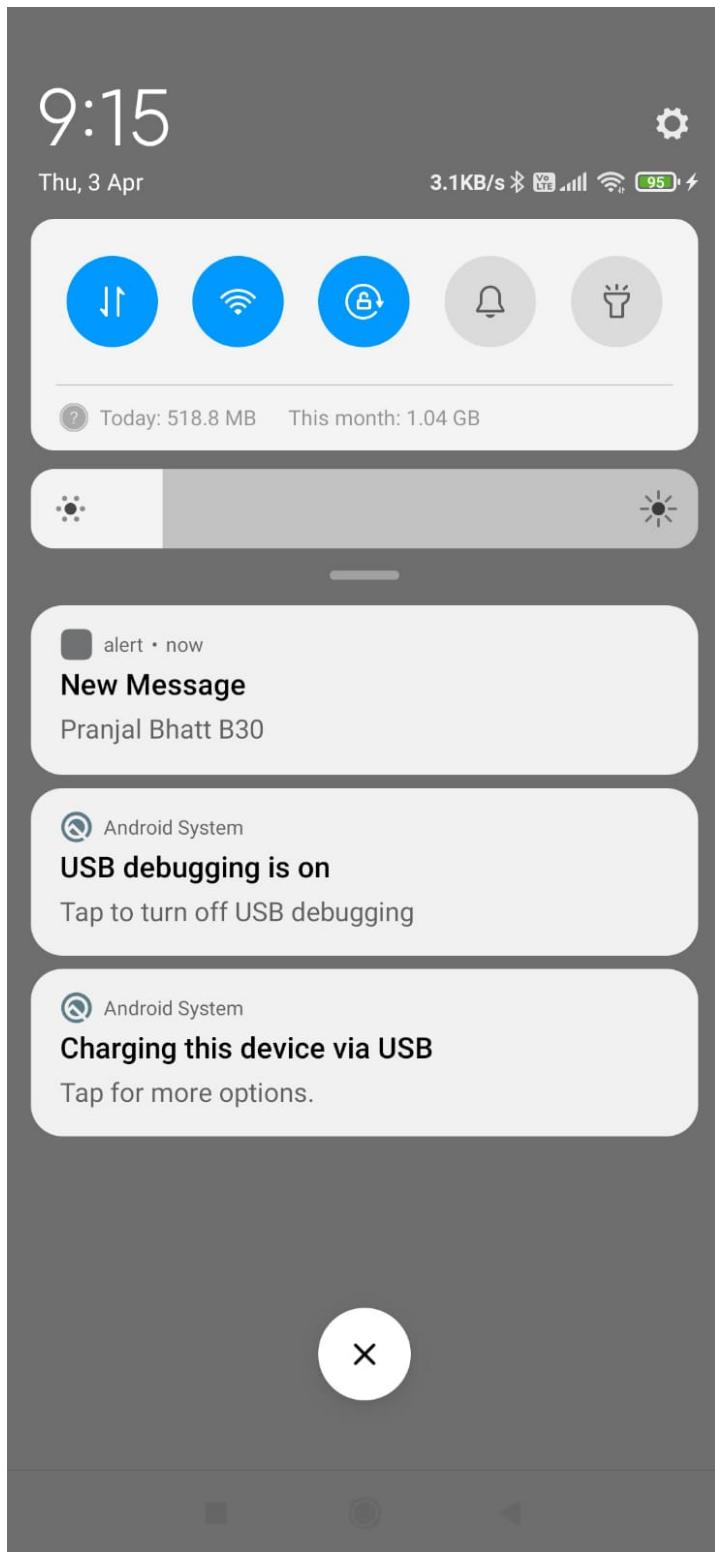
B.2 Input and Output:



Pranjal Bhatt B30

SEND NOTIFICATION





B.3 Observations and learning:

During the experiment, an Android application was developed using Android Studio to create an alert upon receiving a message. The setup involved configuring the Android SDK, JDK, and Gradle, followed by creating a new project. The layout and logic were implemented in XML and Java/Kotlin, respectively. Testing was performed using an emulator or a real device. The application successfully triggered an alert (such as a notification, sound, or pop-up) when a message was received.

B.4 Conclusion:

The experiment demonstrated the practical implementation of Android app development using Android Studio. Students gained hands-on experience with project creation, UI design, and event handling in Android. By working on this alert system, they understood how to integrate background services and notifications into mobile applications.

B.5 Question of Curiosity

1) Explain different steps required to build up this alert project?

- **Set Up Development Environment:**

- • Install Android Studio, Java Development Kit (JDK), and Android SDK.
- Configure Gradle for project dependencies.

- **Create a New Android Project:**

- • Open Android Studio and select "New Project."
- Choose an appropriate project template (e.g., Empty Activity).
- Enter the application name, package name, and select the minimum Android version.
- Click "Finish" to create the project.

- **Design the User Interface (UI):**

- • Navigate to the `res/layout` folder and open `activity_main.xml`.
- Use the graphical layout editor to design the interface (e.g., add a `TextView` to display messages).

- **Implement Message Receiving Functionality:**

- • Modify the `AndroidManifest.xml` file to request the necessary permissions:
- • Register a `BroadcastReceiver` to listen for incoming SMS messages.

- **Write Java/Kotlin Code to Trigger an Alert:**

- • Create a new class extending BroadcastReceiver:

- **Test the Application:**

- • Run the application on an emulator or a physical device.
- Use an SMS testing tool to simulate incoming messages.

- **Debug and Optimize:**

- • Check logs in Logcat for debugging.
- Optimize code for better performance.

Experiment 07: To develop an android application that uses GPS Location Information.

PART A

A.1 Aim: To develop an android application that uses GPS Location Information.

A.2 Objectives: To introduce students with various tools like Android Studio, NS2, Wire-shark, Cisco packet tracer, WAP supported browser etc.

A.3 Outcome: After successful completion of this experiment students will be able to develop an android application that uses GPS Location Information. **A.4 Theory:**

SOFTWARE:

- Android Studio
- The Android SDK (Starter Package)
- Gradle
- Java Development Kit (JDK) 5

DESCRIPTION:

1. Open android studio and select new android project .
2. Give project name and select next
3. Choose the android version.
4. Enter the package name. package name must be two word separated by comma and click finish
5. Go to package explorer in the left hand side and select our project.

6. Go to res folder and select layout. Double click the main.xml file

7. Now you can see the Graphics layout window.

INPUT:

MAINACTIVITY.JAVA

```
package com.gpslocation;

//import android.R; import android.view.View;
import android.widget.Button; import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity; import android.os.Bundle;

public class MainActivity extends AppCompatActivity{

    /** Called when the activity is first created. */ Button
    btnShowLocation; GPStrace gps;

    @Override
    public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnShowLocation=(Button)findViewById(R.id.show_Location);
        btnShowLocation.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub gps=new
                GPStrace(MainActivity.this);
                if(gps.canGetLocation()) { double
                    latitude=gps.getLatitude(); double
                    longitude=gps.getLongitude();
                    Toast.makeText(getApplicationContext(),"Your Location is
                    \nLat:"+latitude+"\nLong:"+longitude,Toast.LENGTH_LONG).show();
                }
                else { gps.showSettingAlert();
                }

            }
        });
    }
}
```

ACTIVITY_MAIN.XML

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

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" android:id="@+id/relativeLayout1" android:layout_width="match_parent" android:layout_height="match_parent">

    <Button android:id="@+id/show_Location" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Show_Location" android:layout_centerVertical="true" android:layout_centerHorizontal="true" />

</RelativeLayout>
```

GPSTRACE.JAVA

```
package com.gpslocation;

import android.app.AlertDialog; import
android.app.Service; import android.content.Context;
import android.content.DialogInterface; import
android.content.Intent; import
android.location.Location; import
android.location.LocationListener; import
android.location.LocationManager; import
android.os.Bundle; import android.os.IBinder;
import android.provider.Settings;

public class GPStrace extends Service implements LocationListener{ private final Context
context; boolean isGPSEnabled=false; boolean canGetLocation=false;
boolean isNetworkEnabled=false; Location location;
double latitude; double longitude;
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES=10; private static
final long MIN_TIME_BW_UPDATES=1000*60*1; protected
```



```
        }
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    return location;
}

public void stopUsingGPS() { if(locationManager!=null) {
    locationManager.removeUpdates(GPStrace.this);
}
}

public      double      getLatitude()      {
    if(location!=null)
        latitude=location.getLatitude();
    }
    return latitude;
}

public      double      getLongtiude()      {
    if(location!=null)
        longitude=location.getLongitude();
    }
    return longitude;
}

public boolean canGetLocation() { return this.canGetLocation;
}

public void showSettingAlert() {
    AlertDialog.Builder alertDialog= new AlertDialog.Builder(context);
}
```

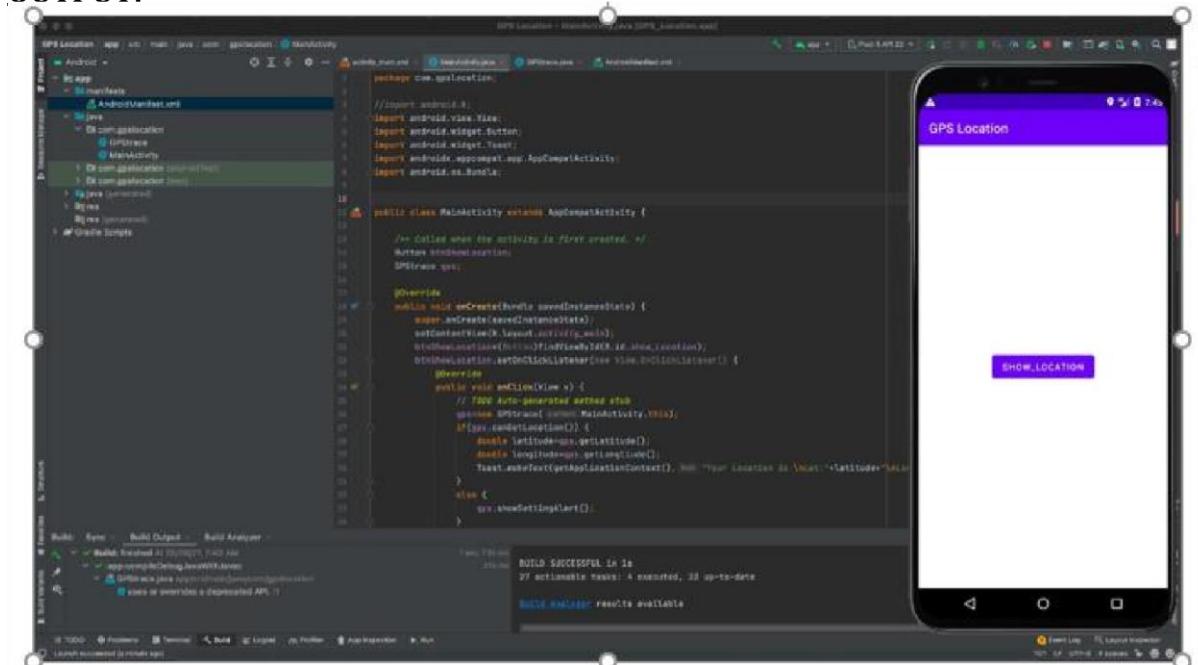
ANDROIDMANIFEST.XML

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.GPSLocation">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

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

OUTPUT:





SHOW_LOCATION

SHOW_LOCATION

Your Location is
Lat:37.42199833333335
Long:37.42199833333335



PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll. No. B30	Name: Pranjal Bhatt
Class :TE COMPS B	Batch:B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student/steps:

MainActivity.java:

```
package com.example.exp7;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.io.IOException;
import java.util.List;
```

```
public class MainActivity extends AppCompatActivity {
    private static final String TAG = MainActivity.class.getSimpleName();
```

```
private TextView latitudeTextView, longitudeTextView;
private EditText editCity;
private Button getLocationButton;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    latitudeTextView = findViewById(R.id.latitudeTextView);
    longitudeTextView = findViewById(R.id.longitudeTextView);
    getLocationButton = findViewById(R.id.getLocationButton);
    editCity = findViewById(R.id.editTextCity);
    getLocationButton.setOnClickListener(v-> getCoordinates());
}
public void getCoordinates() {
    String city = editCity.getText().toString().trim();
    if (city.isEmpty()) {
        Toast.makeText(this, "Please enter a city name", Toast.LENGTH_SHORT).show();
        return;
    }
    new Thread(() -> { // Run in a background thread
        Geocoder geocoder = new Geocoder(this);
        try {
            List<Address> addresses = geocoder.getFromLocationName(city, 1);
            if (addresses == null || addresses.isEmpty()) {
                runOnUiThread(() -> Toast.makeText(this, "Couldn't Fetch Coordinates",
Toast.LENGTH_SHORT).show());
                return;
            }
            Address address = addresses.get(0);
            double latitude = address.getLatitude();
            double longitude = address.getLongitude();

            runOnUiThread(() -> { // Update UI on main thread
                latitudeTextView.setText(String.valueOf(latitude));
                longitudeTextView.setText(String.valueOf(longitude));
            });
        }
    });
}
```

```
        } catch (IOException e) {
            runOnUiThread(() -> Toast.makeText(this, "Error fetching coordinates",
Toast.LENGTH_SHORT).show());
        }
    }).start();
}
```

AndroidManifest.xml:

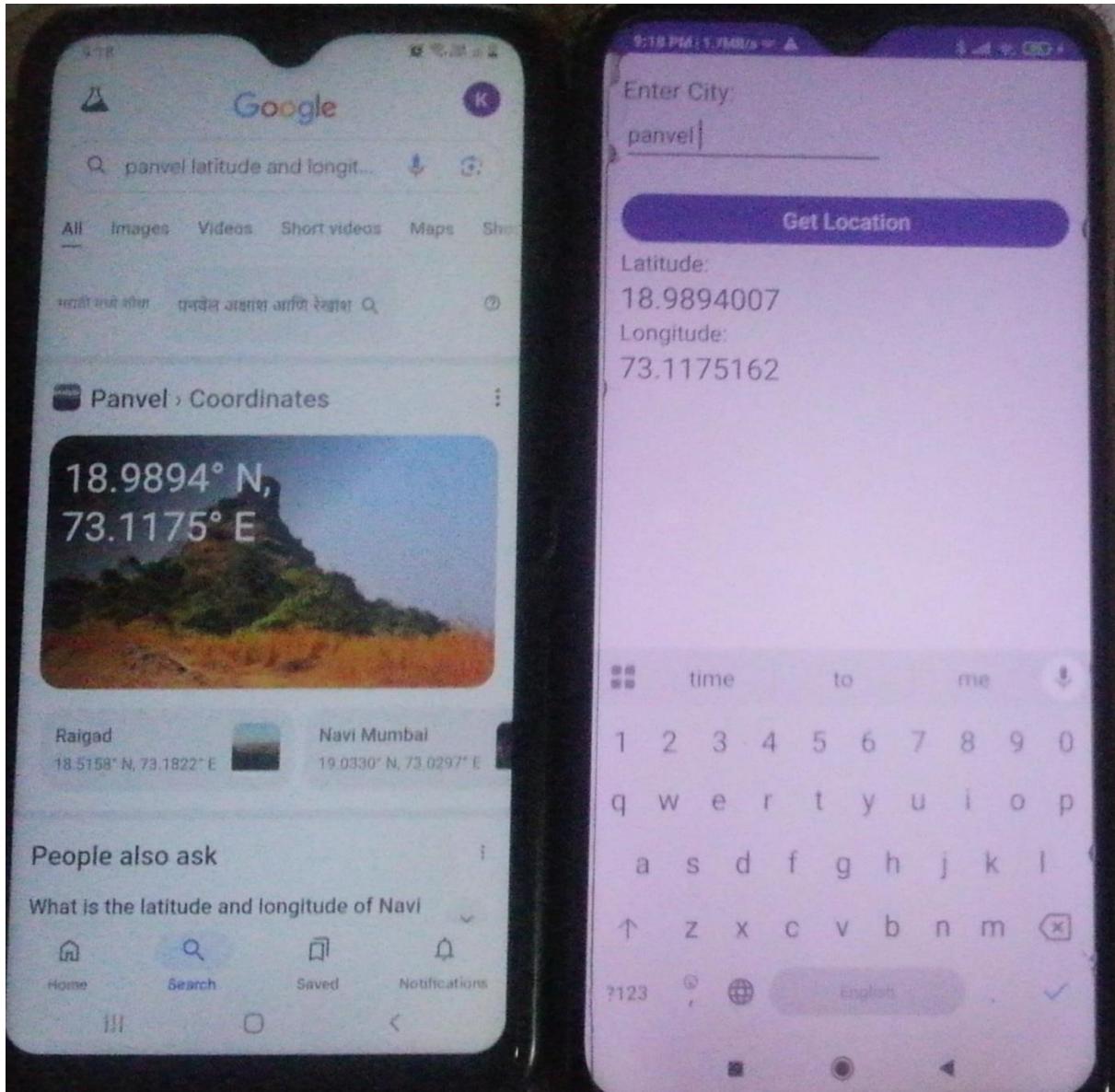
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Exp7"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>

B.2 Input and Output:**B.3 Observations and learning:**

During the experiment, an Android application was developed using Android Studio to fetch and display GPS location information. The setup involved configuring Android Studio, JDK, and Gradle, followed by creating a new project. Necessary permissions for accessing location services were added to the `AndroidManifest.xml` file. The application successfully retrieved latitude and longitude coordinates using the device's GPS sensor and displayed them on the screen. Testing was conducted using an emulator with a simulated location or a physical device with GPS enabled.

B.4 Conclusion:

The experiment demonstrated the practical implementation of GPS-based location tracking in Android applications. Students gained experience with integrating Android's Location API and handling permissions for accessing user location. By completing this project, they understood how real-time location services work and how they can be used in mobile applications for various purposes like navigation, tracking, and location-based services.

B.5 Question of Curiosity

1) Explain different steps required to build up this GPS project?

- **Set Up Development Environment:**

- Install Android Studio, Java Development Kit (JDK), and Android SDK.
 - Configure Gradle dependencies.

- **Create a New Android Project:**

- Open Android Studio and select "New Project."
 - Choose an appropriate project template (e.g., Empty Activity).
 - Enter the application name, package name, and select the minimum Android version.
 - Click "Finish" to create the project.

- **Request Permissions in `AndroidManifest.xml`:**

- **Modify `activity_main.xml` to Display Location:**
- Implement Location Retrieval Logic in Java/Kotlin:
- Handle Permission Requests:

- **Test the Application:**

- Run the application on an emulator with a simulated GPS location.
 - Alternatively, test on a physical device with GPS enabled.

- **Debug and Optimize:**

- Check logs in Logcat for debugging.
 - Improve error handling and UI updates for better user experience.

Experiment 08: Develop an application that uses GUI components, Fonts and colors

PART A

A.1 Aim: To develop an application that uses GUI components, Fonts and colors.

A.2 Objectives: To introduce students with various tools like Android Studio, NS2, Wire-shark, Cisco packet tracer, WAP supported browser etc.

A.3 Outcome:

After successful completion of this experiment students will be able to Develop an application that uses GUI components, Fonts and colours

A.4 Theory:

SOFTWARE:

- Android Studio
- The Android SDK (Starter Package)
- Gradle
- Java Development Kit (JDK) 5

DESCRIPTION:

- 1 Open android studio and select new android project .
- 2 Give project name and select next 3 Choose the android version.

- 4 Enter the package name. package name must be two word separated by comma and click finish
- 5 Go to package explorer in the left hand side and select our project.
- 6 Go to res folder and select layout. Double click the main.xml file 7 Now you can see the Graphics layout window.

Source code:

CodeforActivity_main.xml:

```
<?xmlversion="1.0"encoding="utf-8"?>  
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"android:orient  
ati on="vertical"  
  
    android:layout_width="match_parent"android:layout_height="mat ch_parent">  
  
        <TextVewandroid:id="@+id/textVi  
        ew"android:layout_width="match_  
        parent"  
  
            android:layout_height="wrap_content"  
            a  
            ndroid:layout_margin="30dp"android:  
            g ravity="center"android:text="Hello  
            World!"android:textSize="25sp"androi  
            d:textStyle="bold" />  
  
        <Button  
  
            android:id="@+id/button1"android:lay  
            out_width="match_parent"android:layo
```

```
ut_height="wrap_content" android:layout_margin="20dp" android:gravity="center" android:text="Change font size" android:textSize="25sp"
/>

<Button
    android:id="@+id/button2" android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_margin="20dp" android:gravity="center" android:text="Change color" android:textSize="25sp"
/>

</LinearLayout>
```

CodeforMainActivity.java:

```
ackage com.example.exno1;

import android.graphics.Color;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
```

```
public class MainActivity extends AppCompatActivity
```

```
{  
    intch=1;floa  
    t  
    font=30;@  
    O voverride  
  
    protectedvoidonCreate(Bundle savedInstanceState)  
    { super.onCreate(savedInstanceState);setContentView(R.layout.a ctivity_main); final  
    TextView t= (TextView)  
        findViewById(R.id.textView);Button b1= (Button)  
        findViewById(R.id.button1);b1.setOnClickListener  
        ( new View.OnClickListener(){  
  
            @Override public void  
            onClick(View  
  
                v)  
                {t.setTextSize(font); font  
                 = font +  
  
                  5;if (font  
                  == 50)font  
                  =30;  
                }  
            });  
        Button b2= (Button)  
        findViewById(R.id.button2);b2.setOnClickListener  
        (new View.OnClickListener(){  
  
            @Override
```

```
public void onClick(View  
v)  
  
{switch(ch){ case1:  
  
    t.setTextColor(Color.RED); break  
  
 case2:  
    t.setTextColor(Color.GREEN); break;  
  
 case3:  
    t.setTextColor(Color.BLUE)  
    ; break;  
  
 case4:  
    t.setTextColor(Color.CYAN); break;  
  
 case5:  
    t.setTextColor(Color.YELLOW); break;
```



```
case6: t.setTextColor(Color.MAGENTA);
        break;
    }
}
});
}
}
ch++;
if ( ch == 7 ): ch
    == 1;
```

Output:



PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll. No. B30	Name: Pranjal Bhatt
Class :TE COMPS B	Batch:B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student/steps:

MainActivity.java:

```
package com.example.exno8;
```

```
import android.content.SharedPreferences;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatDelegate;
```

```
public class MainActivity extends AppCompatActivity {
    private int colorIndex = 0;
    private float fontSize = 30f;
```

```
private final int[] colors = {Color.RED, Color.GREEN, Color.BLUE, Color.CYAN, Color.YELLOW, Color.BLACK};

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

    // Load theme preference
    SharedPreferences preferences = getSharedPreferences("AppPrefs", MODE_PRIVATE);
    boolean isDarkMode = preferences.getBoolean("DarkMode", false);
    AppCompatDelegate.setDefaultNightMode(isDarkMode ?
        AppCompatDelegate.MODE_NIGHT_YES : AppCompatDelegate.MODE_NIGHT_NO);

    setContentView(R.layout.activity_main);

    // Initialize views
    TextView textView = findViewById(R.id.textView);
    Button buttonSize = findViewById(R.id.button1);
    Button buttonColor = findViewById(R.id.button2);
    Button buttonTheme = findViewById(R.id.button3);

    // Increase font size on button click
    buttonSize.setOnClickListener(v -> {
        fontSize += 5;
        if (fontSize > 50) {
            fontSize = 30;
        }
        textView.setTextSize(fontSize);
    });

    // Change text color on button click
    buttonColor.setOnClickListener(v -> {
        textView.setTextColor(colors[colorIndex]);
        colorIndex = (colorIndex + 1) % colors.length;
    });

    // Toggle Dark Mode
    buttonTheme.setOnClickListener(v -> {
```

```
        boolean      isDark      =      AppCompatDelegate.getDefaultNightMode()      ==  
AppCompatDelegate.MODE_NIGHT_YES;  
        AppCompatDelegate.setDefaultNightMode(isDark ?  
            AppCompatDelegate.MODE_NIGHT_NO : AppCompatDelegate.MODE_NIGHT_YES);  
  
        // Save preference  
        SharedPreferences.Editor editor = preferences.edit();  
        editor.putBoolean("DarkMode", !isDark);  
        editor.apply();  
  
        // Restart activity to apply theme change  
        recreate();
```

AndroidManifest.xml:

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Exno8"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</application>

</manifest>

B.2 Input and Output:

9:20 PM | 1.5KB/s ▲ ▲ ⚡ 4G LTE 95% 🔋

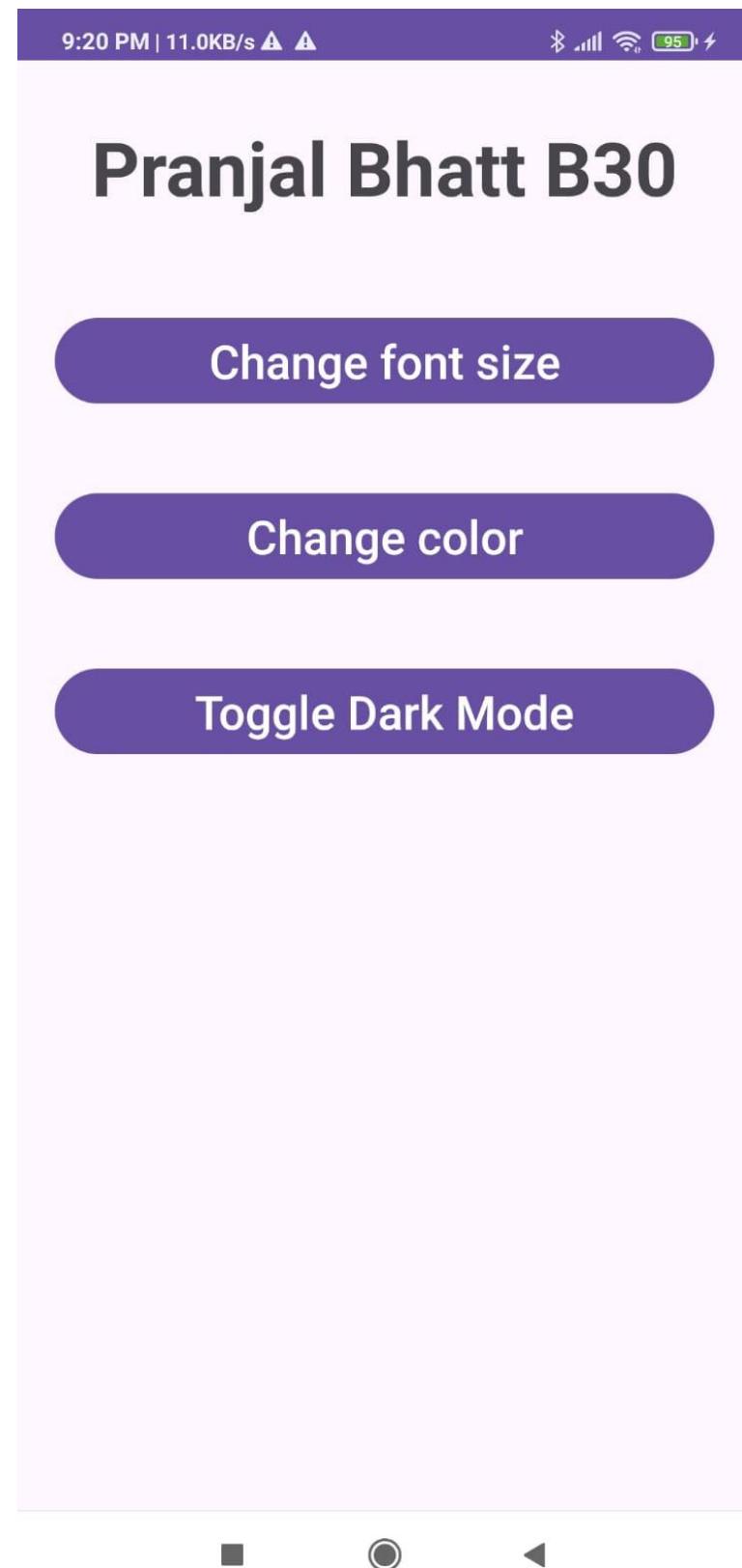
Pranjal Bhatt B30

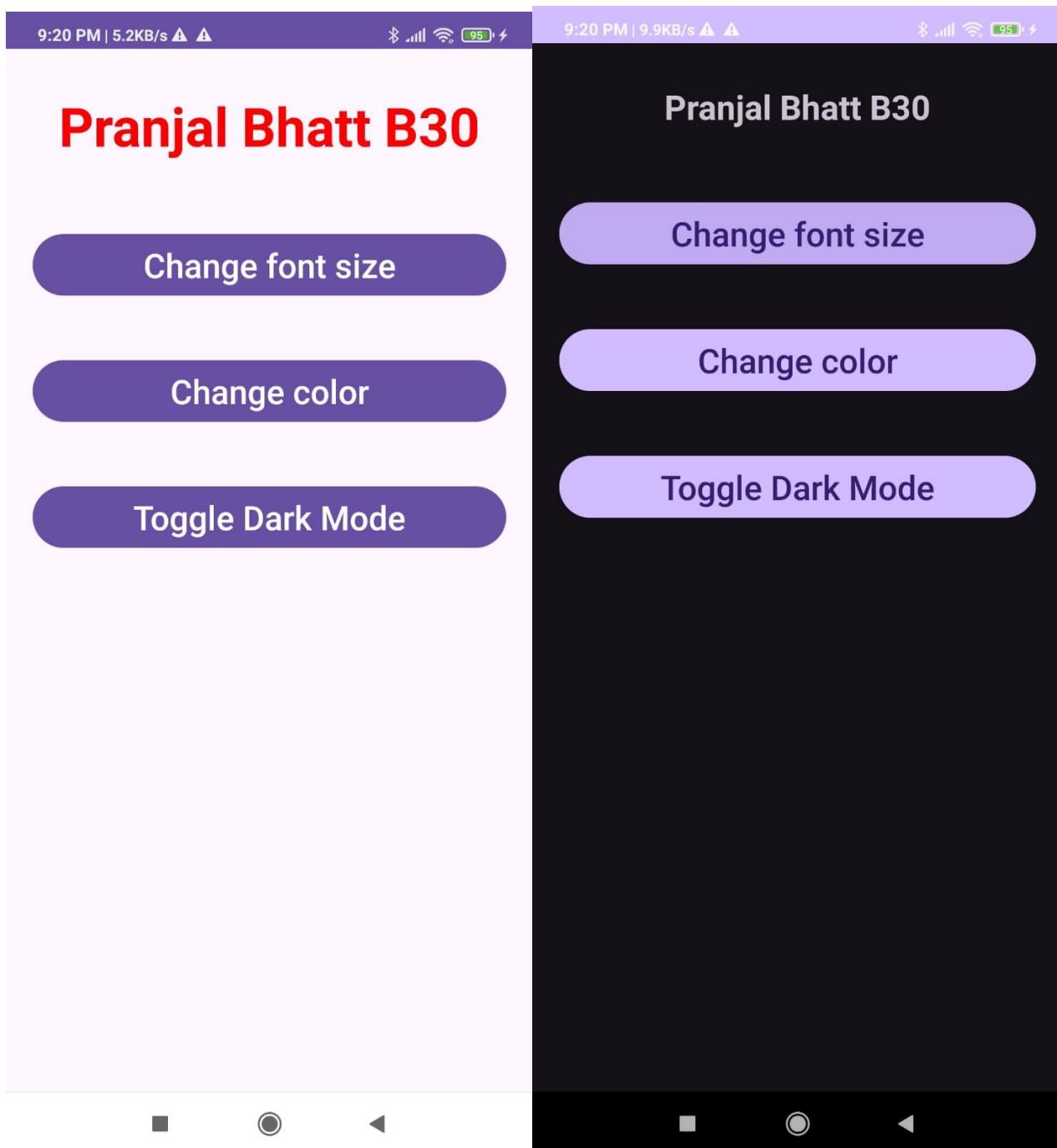
Change font size

Change color

Toggle Dark Mode







B.3 Observations and learning:

During the experiment, an Android application was developed using Android Studio to implement GUI components, fonts, and colors. Various UI elements like TextView, Button, EditText, and ImageView were added and customized using different fonts and colors. XML and Java/Kotlin code were used to define the layout and functionality. The project successfully displayed a user interface with custom-styled

text, buttons, and background colors. Testing was conducted on an emulator and a physical device to verify proper rendering of the UI components.

B.4 Conclusion:

The experiment provided hands-on experience in designing Android user interfaces with GUI components, fonts, and colors. Students learned how to customize UI elements using XML properties and Java/Kotlin code. This experiment helped them understand the importance of UI design in application development and how to enhance the visual appeal of an Android application.

B.5 Question of Curiosity

- 1) Explain different steps required to build up this project?

Steps Required to Build the Project:

Set Up Development Environment:

1. Install Android Studio, Java Development Kit (JDK), and Android SDK.
2. Configure Gradle dependencies.

Create a New Android Project:

1. Open Android Studio and select "New Project."
2. Choose an appropriate project template (e.g., Empty Activity).
3. Enter the application name, package name, and select the minimum Android version.
4. Click "Finish" to create the project.

Modify `activity_main.xml` to Add GUI Components:

1. Navigate to `res/layout/activity_main.xml`.
2. Add UI components like `TextView`, `EditText`, `Button`, and `ImageView`.

Customize Fonts and Colors:

- Define custom fonts by placing `.ttf` files inside `res/font/`.
- Apply the custom font in `TextView`

Write Java/Kotlin Code to Handle Button Clicks:

- Open `MainActivity.java` and modify button functionality:

Run and Test the Application:

- • Launch the emulator or use a physical device.
- Verify that UI components display correctly.
- Test button functionality and color changes.

Debug and Optimize:

- • Check logs in Logcat for debugging.
- Adjust font sizes and colors for better readability.

Experiment 09: Write an application that draws basic graphical primitives in the screen

PART A

A.1 Aim: To write an application that draws basic graphical primitives in the screen.

A.2 Objectives: To introduce students with various tools like Android Studio, NS2, Wire-shark, Cisco packet tracer, WAP supported browser etc.

A.3 Outcome: After successful completion of this experiment students will be able to develop an application that draws basic graphical primitives in the screen.

A.4 Theory:

SOFTWARE:

- Android Studio
- The Android SDK (Starter Package)
- Gradle
- Java Development Kit (JDK) 5

DESCRIPTION:

8. Open android studio and select new android project .
9. Give project name and select next
10. Choose the android version.

11. Enter the package name. package name must be two word separated by comma and click finish
12. Go to package explorer in the left hand side and select our project.
13. Go to res folder and select layout. Double click the main.xml file 14. Now you can see the Graphics layout window.

SourceCode:

CodeforActivity_main.xml:

```
<?xmlversion="1.0"encoding="utf-8"?>

<RelativeLayoutxmlns:android="http://schemas.android.com/apk
    /res/android"android:layout_width="match_parent"android:layo
    ut_height=" match_parent">

    <ImageView
        android:layout_width="match_pare
        nt"android:layout_height="match_p
        arent"android:id="@+id/imageVie
        w"/>

    < /RelativeLayout >
```

Code for MainActivity.java:

```
package com.example.exno4;
import android.app.Activity;
import
android.graphics.Bitmap;
import
android.graphics.Canvas;

import
android.graphics.Color;
import android.graphics.Pain
t;

import android.graphics.drawable.BitmapDrawable;

import android.os.Bundle;
import android.widget.ImageView;
public class MainActivity extends Activi
ty

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

        //Creating a Bitmap
        Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_888
8);

        //Setting the Bitmap as background for the
        ImageView imageView = (ImageView)
```

```
findViewById(R.id.imageView);i.setBackgroundDraw ab
le(newBitmapDrawable(bg)); //CreatingtheCanvasObject

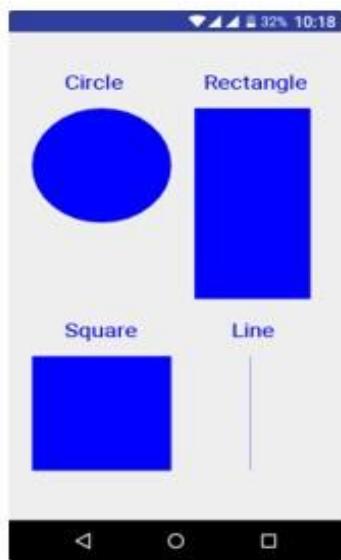
Canvascanvas=newCanvas(bg
);
//Creating the Paint Object and set its color
&TextSizePaint paint = new
Paint();paint.setColor(Color.BLUE);paint.se
tTextSize(50);

//To draw a

        Rectanglecanvas.drawText("Rectangle",
420,150,paint); canvas.drawRect(400,200,650,700,pain
t);
```

```
//Todrawa Circle  
    canvas.drawText("Circle",120,150,paint);  
    canvas.drawCircle(200,350,150,paint);  
  
//Todrawa Square  
    canvas.drawText("Square",120,800,paint);  
    canvas.drawRect(50,850,350,1150,paint);  
  
//Todrawa Line  
    canvas.drawText("Line",480,800,paint);  
  
    canvas.drawLine(520,850,520,1150,paint);  
}  
}
```

Output:



PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll. No. B30	Name: Pranjal Bhatt
Class TE B Comps	Batch: B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student/steps:

MainActivity.java:

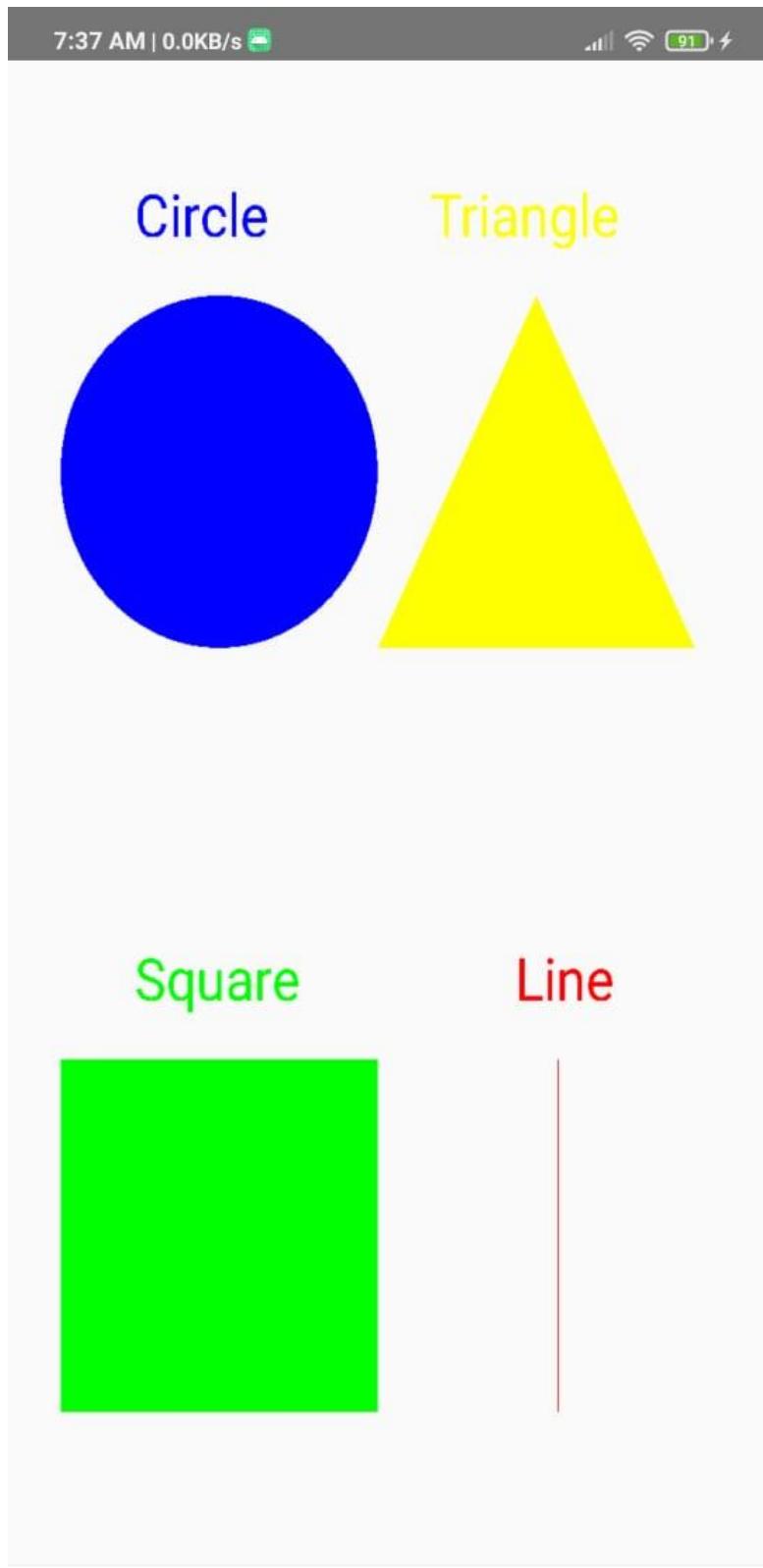
```
package com.example.exp9;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Path;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.widget.ImageView;
public class MainActivity extends Activity {
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main); // Make sure this matches the XML file name
    // Creating a Bitmap
    Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888); // Setting the
    Bitmap as background for the ImageView
    ImageView i = findViewById(R.id.imageView); // Ensure ID matches XML
    i.setBackground(new BitmapDrawable(getResources(), bg)); // Updated deprecated
    method
    // Creating the Canvas Object
    Canvas canvas = new Canvas(bg);
    // Creating the Paint Object and setting its color & TextSize
    Paint paint = new Paint();
    paint.setColor(Color.BLUE);
    paint.setTextSize(50);
    // Drawing shapes
    paint.setColor(Color.BLUE);
    canvas.drawText("Circle", 120, 150, paint);
    canvas.drawCircle(200, 350, 150, paint);
    paint.setColor(Color.GREEN);
    canvas.drawText("Square", 120, 800, paint);
    canvas.drawRect(50, 850, 350, 1150, paint);
    paint.setColor(Color.YELLOW);
    canvas.drawText("Triangle", 400, 150, paint);
    Path path = new Path();
    path.moveTo(500, 200);
    path.lineTo(350, 500);
    path.lineTo(650, 500);
    path.close();
    canvas.drawPath(path, paint);
    paint.setColor(Color.RED);
    canvas.drawText("Line", 480, 800, paint);
    canvas.drawLine(520, 850, 520, 1150, paint);
}
}
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.exp9">
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.EXP9">
    <activity
        android:name=".MainActivity"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
</manifest>
```

B.2 Input and Output:



B.3 Observations and learning:

During the experiment, an application was successfully developed using Android Studio to draw basic graphical primitives such as rectangles, circles, squares, and lines on the screen. The implementation involved setting up an Android project, defining the layout in activity_main.xml, and writing Java code in MainActivity.java to create a Canvas object and use the Paint class to draw shapes. The experiment allowed students to understand how graphical elements are rendered in an Android application. Additionally, the experiment introduced students to essential development tools like Android Studio, Gradle, and the Android SDK, helping them gain hands-on experience in mobile application development.

B.4 Conclusion:

This experiment provided practical exposure to developing a simple graphics-based Android application, enhancing students' understanding of GUI programming in mobile applications. By using Android Studio and Java, students learned how to work with the Canvas and Paint classes to draw and manipulate graphical objects dynamically. The experiment also reinforced the importance of object-oriented programming concepts in application development. Overall, it served as a foundation for more advanced graphics programming in Android, preparing students for future projects involving interactive user interfaces and custom graphics.

B.5 Question of Curiosity

1) Explain different steps required to build up this project?

1. Open android studio and select new android project.
2. Give project name and select next
3. Then select the Minimum SDK as shown below and click Next.
4. Then select the Empty Activity and click Next.
5. Finally click Finish.
6. Click on app -> res -> layout -> activity_main.xml.
7. Drag and drop relative layout and change its propertiesDrag and drop image view andchange its properties according to our programsScreen layout can be viewed by clicking graphics layout tab
8. Now click on Design and your application will look as given below.
9. Click on app -> java -> com.example.exp4 -> MainActivity.
10. Override OnCreate() function
11. Create bitmap and canvas objects.

12. Save the program.
13. Run the program.
14. Output can be viewed in the android emulator.

Experiment 10: Implementation of income tax/loan EMI calculator and deploy the same on real devices

PART A

A.1 Aim: To Implementation of income tax/loan EMI calculator and deploy the same on real devices.

A.2 Objectives: To introduce students with various tools like Android Studio, NS2, Wireshark, Cisco packet tracer, WAP supported browser etc.

A.3 Outcome: After successful completion of this experiment students will be able to Implementation of income tax/loan EMI calculator and deploy the same on real devices

A.4 Theory:

SOFTWARE:

- Android Studio
- The Android SDK (Starter Package)
- Gradle
- Java Development Kit (JDK) 5

DESCRIPTION:

15. Open android studio and select new android project .
16. Give project name and select next

17. Choose the android version.
18. Enter the package name. package name must be two word separated by comma and click finish
19. Go to package explorer in the left hand side and select our project.
20. Go to res folder and select layout. Double click the main.xml file 21. Now you can see the Graphics layout window.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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" android:background="#D8D0D0"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="8dp" android:ems="10"
        android:hint="@string/enter_principle"
        android:inputType="number"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.45"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.203" android:autofillHints=""
    />

    <EditText android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:ems="10"
        android:hint="@string/enter_time"
        android:inputType="number"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.45"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.274"
        android:autofillHints="" />
```

```
<EditText android:id="@+id/editText3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:ems="10"
    android:hint="@string/enter_rate_of_interest" android:inputType="number"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.45"
    app:layout_constraintStart_toStartOf="parent"
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:text="@string/calculate_emi"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.62" />

<TextView
    android:id="@+id/textView"
    android:layout_width="95dp"
    android:layout_height="104dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.486"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.901" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.emicalculator;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

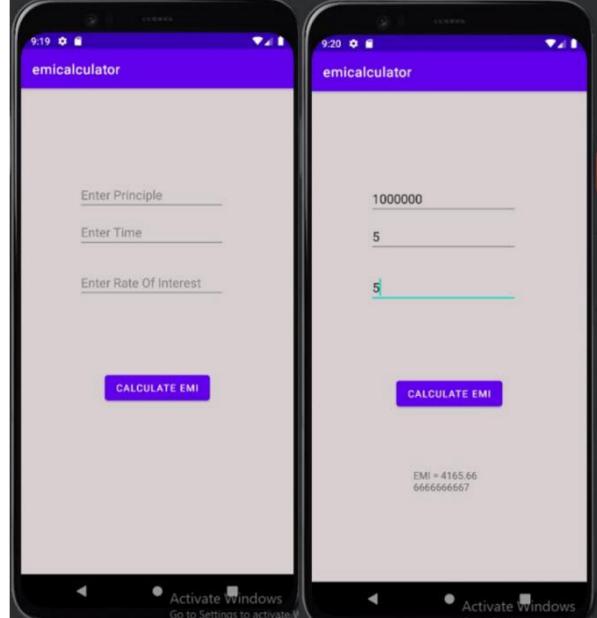
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    Button calculate;
    TextView result;
    EditText pe, ne, re;

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

        calculate = findViewById(R.id.button);
        result = findViewById(R.id.textView);
        pe = findViewById(R.id.editText);
        ne = findViewById(R.id.editText2);
        re = findViewById(R.id.editText3);
        calculate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Double p = Double.parseDouble(pe.getText().toString());
                Double n = Double.parseDouble(ne.getText().toString());
                Double r = Double.parseDouble(re.getText().toString());
```

```
        Double rate= (r/1200);  
        Double res = (p * rate * Math.pow(1 + rate, n) / Math.pow(1 + rate, n) - 1);  
        result.setText("EMI = "+String.valueOf(res));  
    }  
}  
}  
}
```

Output

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll. No. B30	Name: Pranjal Bhatt
Class :TE B Comps	Batch: B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student/steps:

MainActivity.java:

```
package com.example.exp10;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
```

```
import java.text.DecimalFormat;
public class MainActivity extends AppCompatActivity {
private EditText loanAmountEditText;
private EditText interestRateEditText;
private EditText loanTenureEditText;
private TextView resultTextView;
@SuppressWarnings("SetTextI18n")
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
loanAmountEditText = findViewById(R.id.loanAmountEditText);
interestRateEditText = findViewById(R.id.interestRateEditText);
loanTenureEditText = findViewById(R.id.loanTenureEditText);
Button calculateButton = findViewById(R.id.calculateButton);
resultTextView = findViewById(R.id.resultTextView);
calculateButton.setOnClickListener(view -> {
try {
// Read input values
double loanAmount =
Double.parseDouble(loanAmountEditText.getText().toString().trim());
double interestRate =
Double.parseDouble(interestRateEditText.getText().toString().trim());
int loanTenure = Integer.parseInt(loanTenureEditText.getText().toString().trim());
// Validate input values
if (loanAmount <= 0 || loanTenure <= 0) {
resultTextView.setText("Please enter valid loan amount and tenure.");
return;
}
double monthlyInterestRate = interestRate / 1200;
int numberOfPayments = loanTenure * 12;
double emi;
if (interestRate == 0) {
// Simple division if no interest
emi = loanAmount / numberOfPayments;
} else {
```

```
// EMI formula
emi = (loanAmount * monthlyInterestRate * Math.pow(1 + monthlyInterestRate,
numberofPayments))
/ (Math.pow(1 + monthlyInterestRate, numberofPayments) - 1);
}
DecimalFormat decimalFormat = new DecimalFormat("#.###");
resultTextView.setText("EMI: " + decimalFormat.format(emi));
} catch (NumberFormatException e) {
resultTextView.setText("Invalid input! Please enter valid numbers.");
}
});
}
}
}

AndroidManifest.xml:
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Exp10"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

</manifest>

B.2 Input and Output:

9:29 PM | 15.2KB/s ▶️ 🔍 95% ⚡

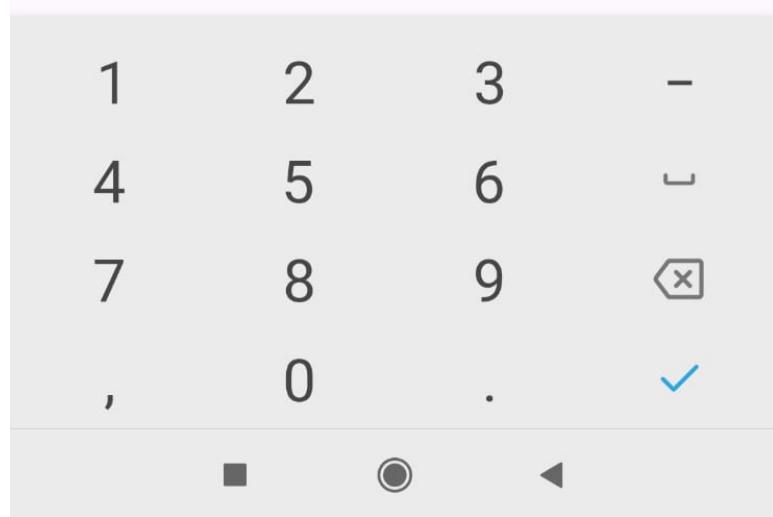
100000000

12

15

Calculate

EMI: 1200168.06



B.3 Observations and learning:

In this experiment, an Android application was successfully developed to calculate loan EMI based on user inputs for principal amount, interest rate, and loan tenure. The implementation used Android Studio, Java, and XML to design the user interface and perform EMI calculations. The app included input fields for entering financial data, a button to trigger the calculation, and a TextView to display the EMI result. The experiment demonstrated how to handle user input, perform mathematical computations using Java, and update UI elements dynamically. Additionally, the application was tested on real devices to ensure proper functionality.

B.4 Conclusion:

This experiment provided valuable hands-on experience in developing a functional Android application for financial calculations. By working with UI components, event handling, and mathematical operations, students gained practical knowledge of mobile app development. Deploying the application on real devices helped in understanding real-world constraints such as screen responsiveness and user experience. The experiment reinforced the importance of software tools like Android Studio and Java for developing interactive applications. This knowledge can be extended to build more complex financial applications, making it a useful foundation for real-world software development.

B.5 Question of Curiosity

Explain different steps required to build up this project?

1. Create a new project in Android Studio with an empty activity.
2. Open the activity_main.xml file and design the user interface for the loan calculator. This can include EditText views for the loan amount, interest rate, and loan term, as well as a TextView for the calculated monthly payment.
3. In the MainActivity.java file, declare variables for the loan amount, interest rate, loan term, and monthly payment. Also, create references to the EditText and TextView views.
4. Implement a listener for the calculate button that extracts the values from the EditText views, calculates the monthly payment using the loan formula, and displays the result in the TextView.
$$\text{Monthly payment} = (\text{loan amount} * \text{interest rate} / 1200) / (1 - (1 + \text{interest rate} / 1200)^{-\text{loan term}})$$
5. Use the following formula to calculate the monthly payment:
$$\text{Monthly payment} = (\text{loan amount} * \text{interest rate} / 1200) / (1 - (1 + \text{interest rate} / 1200)^{-\text{loan term}})$$
6. Run the app to test the loan calculator

