



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

**T.E/SEM V/CBCGS/AIML  
Academic Year: 2022-23**

<b>NAME</b>	<b>SINGH SUDHAM DHARMENDRA</b>
<b>BRANCH</b>	<b>CSE-(AI&amp;ML)</b>
<b>ROLL NO.</b>	<b>AIML57</b>
<b>SUBJECT</b>	<b>WEB COMPUTING AND NETWORK LAB</b>
<b>COURSE CODE</b>	<b>CSL501</b>
<b>PRACTICAL NO.</b>	
<b>DOP</b>	
<b>DOS</b>	

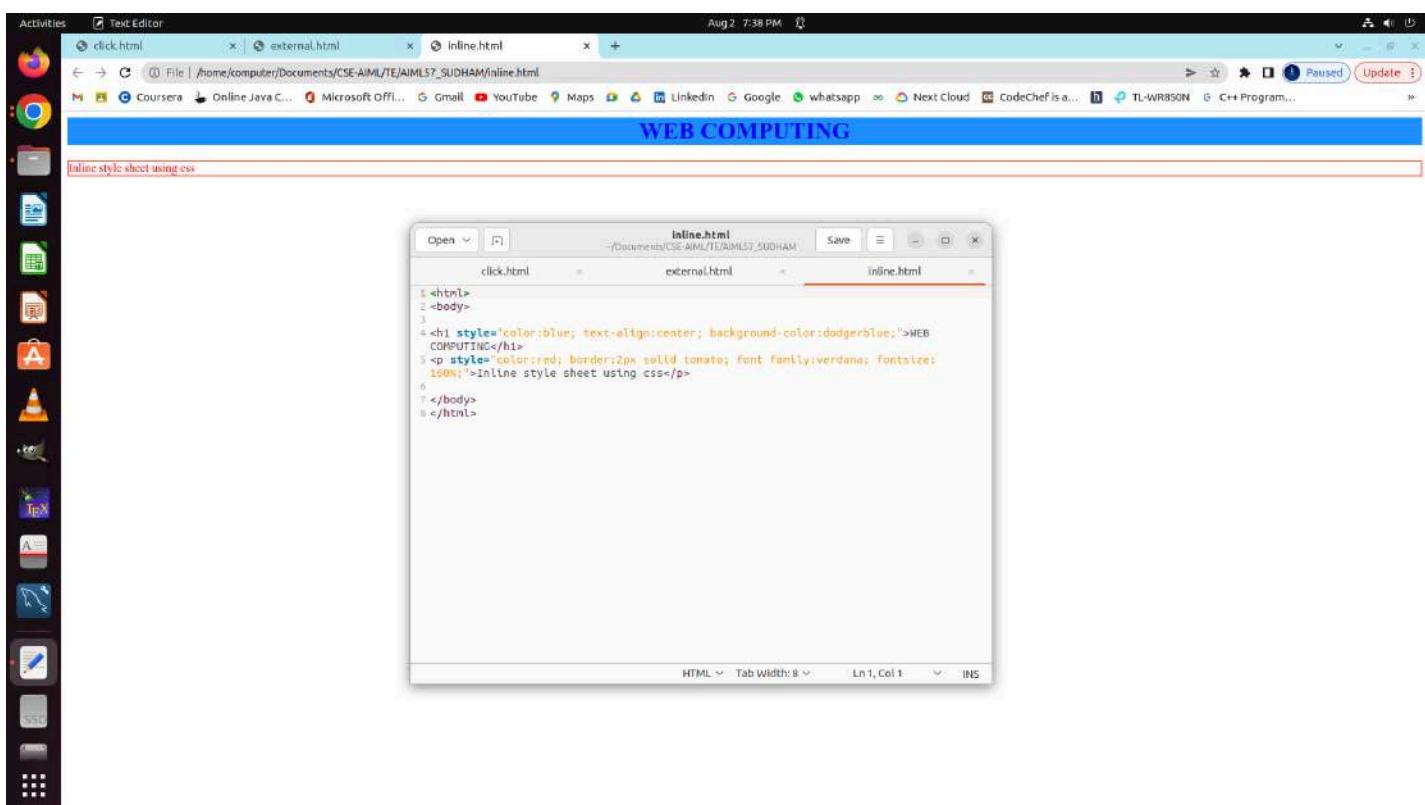


## Code -

### 1.inline.html

```
<html>
<body>
<h1 style="color:blue; text-align:center; background-color:dodgerblue;">WEB
COMPUTING</h1>
<p style="color:red; border:2px solid tomato; font family:verdana;
fontsize:160%;">Inline style sheet using css</p>
</body>
</html>
```

### Output (ScreenShot) -



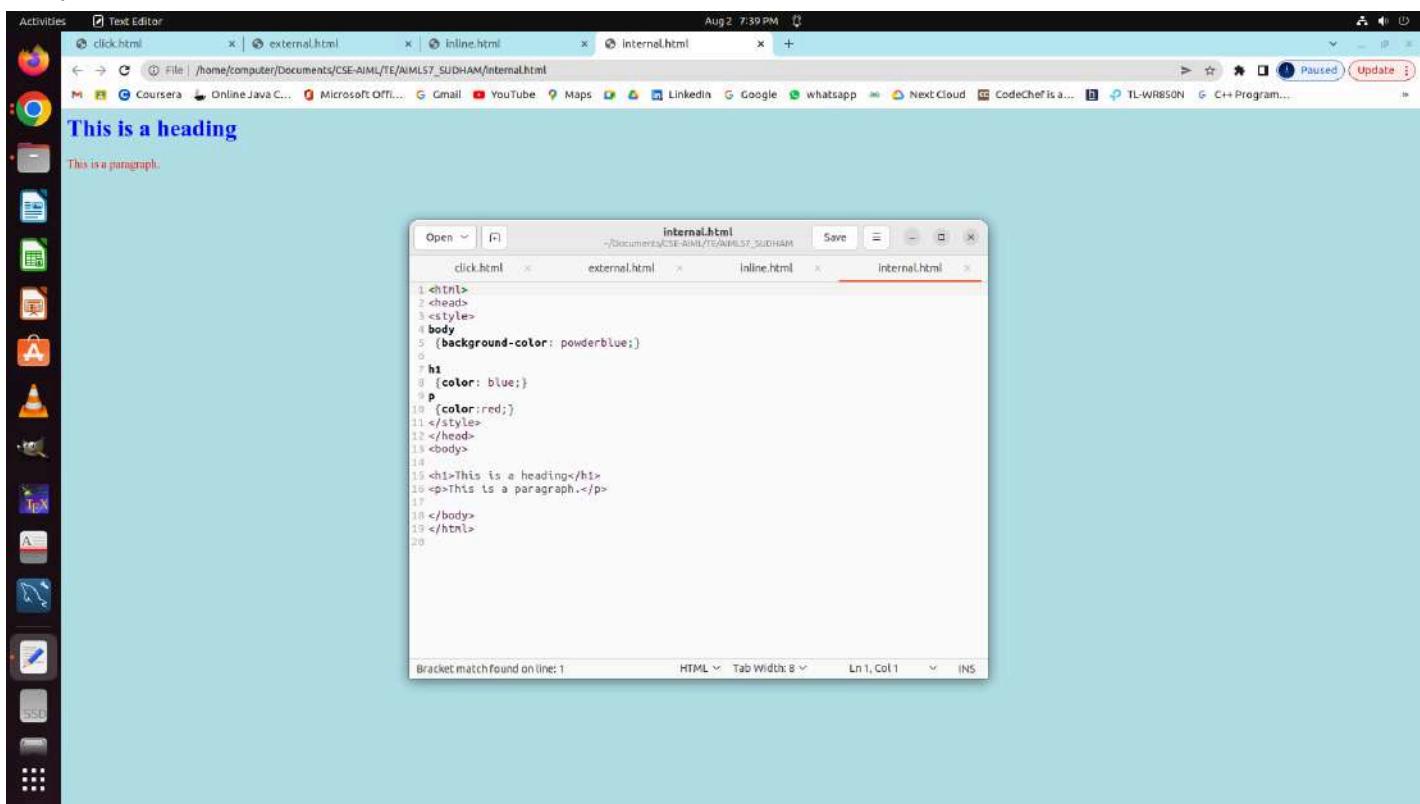
### 2.internal.html

```
<html>
<head>
<style>
body
```



```
{background-color: powderblue;}  
h1  
{color: blue;}  
p  
{color:red;}  
</style>  
</head>  
<body>  
<h1>This is heading</h1>  
<p>This is paragraph</p>  
</body>  
</html>
```

### Output (ScreenShot) -



### 3. external.html

```
<html>  
<head>
```



```
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
<h1>WEB COMPUTING</h1>
<p>EXTERNAL STYLE SHEET USING CSS</p>
</body>
</html>
```

### mystyle.css

```
body
{background-color:powderblue;}
h1
{color:blue;}
```

```
p
{color:red;}
```

### Output (ScreenShot) -

The screenshot shows a desktop environment with three windows open:

- Browser Window:** Displays the output of the HTML file. The title is "WEB COMPUTING" and the content is "EXTERNAL STYLE SHEET USING CSS".
- HTML Editor Window:** Shows the source code of the HTML file. It includes a warning message: "This file "/home/computer/Documents...\_ML57\_SUDHAM/internal.html" is already open in another window. Do you want to edit it anyway?". The code is:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>
<h1>WEB COMPUTING</h1>
<p>EXTERNAL STYLE SHEET USING CSS</p>
</body>
</html>
```
- CSS Editor Window:** Shows the content of the mystyle.css file. The code is:

```
body
{background-color:powderblue;}
h1
{color:blue;}
p
{color:red;}
```



#### 4.click.html

```
<html>
<head>
<style>
body
{background-color: powderblue;}
h1
{color: blue;}
p
{color:red;}
a:link {color:red;}
a:visited {color:green;}
a:hover {color:hotpink;}
a:active {color:blue;}
</style>
</head>
<body>
<p class="intro">This is a paragraph.</p>
<h2>styling link depends on state</h2>
<p><a href="http://www.google.com/" target="_blank">This is a link</a></p>
</body>
</html>
```



## Output (ScreenShot) -

This is a paragraph.

## styling link depends on state

This is a link

```
1 <html>
2   <head>
3     <style>
4       body {
5         background-color: powderblue;
6         h1 {
7           color: blue;
8         }
9         p {
10           color:red;
11         }
12         a:link {color:red;}
13         a:visited {color:green;}
14         a:hover {color:hotpink;}
15         a:active {color:blue;}
16       </style>
17     </head>
18     <body>
19       <p class="intro">This is a paragraph.</p>
20
21       <h2>styling link depends on state</h2>
22       <p><a href="http://www.google.com/" target="_blank">This is a link</a></p>
23
24
25
26
27
28   </body>
29 </html>
30
```

The screenshot shows a Linux desktop environment with a terminal window at the top and several application windows below.

**Terminal:** The terminal window displays a Google search results page for "internal.html". The results show the following snippet from internal.html:

```
1<html>
2 <head>
3 <style>
4   body {
5     background-color: powderblue;
6   }
7   h1 {
8     color: blue;
9   }
10  p {
11    color:red;
12  }
13  a:link {color:red;}
14  a:visited {color:green;}
15  a:hover {color:hotpink;}
16  a:active {color:blue;}
17 </style>
18 </head>
19 <body>
20 <p class="intro">This is a paragraph.</p>
21 <h2>styling link depends on states</h2>
22 <p><a href="http://www.google.com/" target="_blank">This is a
23 link</a></p>
24
25
26
27
28 </body>
29 </html>
```

**Code Editor:** A code editor window titled "click.html" is open, showing the same HTML code as the terminal. The code includes CSS styles for the body, h1, p, and a elements, and an anchor tag with a href attribute pointing to "http://www.google.com/" and a target attribute set to "\_blank".



## 5.table.html

```
<html>
<head>
<style>
table,th,td{border:1px solid black;}
table{border-spacing:15px;}
th,td {padding:25px;}
</style>
</head>
<body>
<h2>INFORMATRION TABLE:</h2>
<table>
<tr>
<th>Sr.No.</th>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
</tr>
<tr>
<td>1</td>
<td>PRATHAMESH</td>
<td>CHIKANKAR</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>SUDHAM</td>
<td>SINGH</td>
<td>19</td>
</tr>
</table>
</body>
</html>
```



## Output (ScreenShot) -

Aug 2 7:50 PM

File /home/computer/Documents/CSE-AIML/TE/AIML57\_SUDHAM/qwe/table.html

Coursera Online Java C... Microsoft Off... Gmail YouTube Maps LinkedIn Google WhatsApp Next Cloud CodeChef... TL-WR850N C++ Program...

INFORMATRION TABLE:

Sr.No.	Firstname	Lastname	Age
1	PRATHAMESH	CHIKANKAR	19
2	SUDHAM	SINGH	19

table.html

```
<html>
<head>
<style>
table,td{border:1px solid black;}
table{border-spacing:15px;}
th,td {padding:25px;}
</style>
</head>
<body>
<h2>INFORMATRION TABLE:</h2>
<table>
<tr>
<th>Sr.No.</th>
<th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
</tr>
<tr>
<td>1</td>
<td>PRATHAMESH</td>
<td>CHIKANKAR</td>
<td>19</td>
</tr>
<tr>
<td>2</td>
<td>SUDHAM</td>
<td>SINGH</td>
<td>19</td>
</tr>
</table>
</body>
</html>
```

HTML Tab Width: 8 Ln 1, Col 1 INS



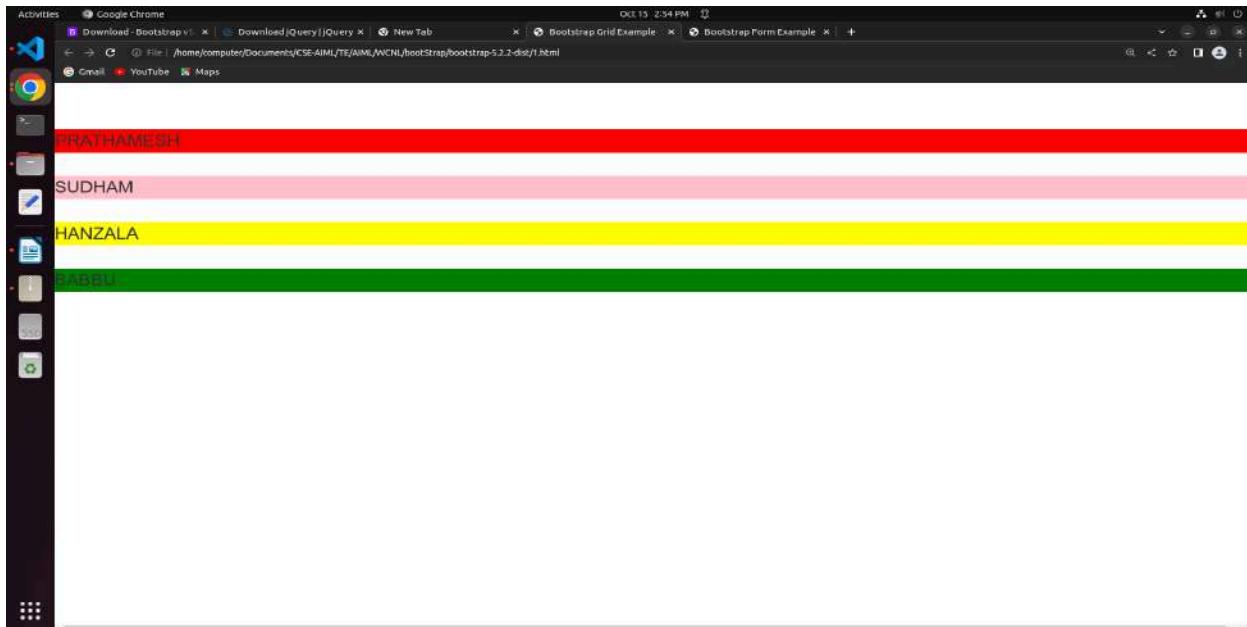
## **EXPERIMENT NO.03 - BOOTSTRAP**

### **1.Bootstrap Grid systems**

**CODE:**

```
<html>
<head>
    <title>Bootstrap Grid Example</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script
        src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>
    <div class="row">
        <div class="col-md-3" style="background-color:red;">PRATHAMESH</div>
        <div class="col-md-3" style="background-color:pink;">SUDHAM</div>
        <div class="col-md-3" style="background-color:yellow;">HANZALA</div>
        <div class="col-md-3" style="background-color:green;">BABBHU</div>
    </div>
</body>
</html>
```

**OUTPUT:**





## 2.Bootstrap vertical form

**CODE:**

**formStyle.css**

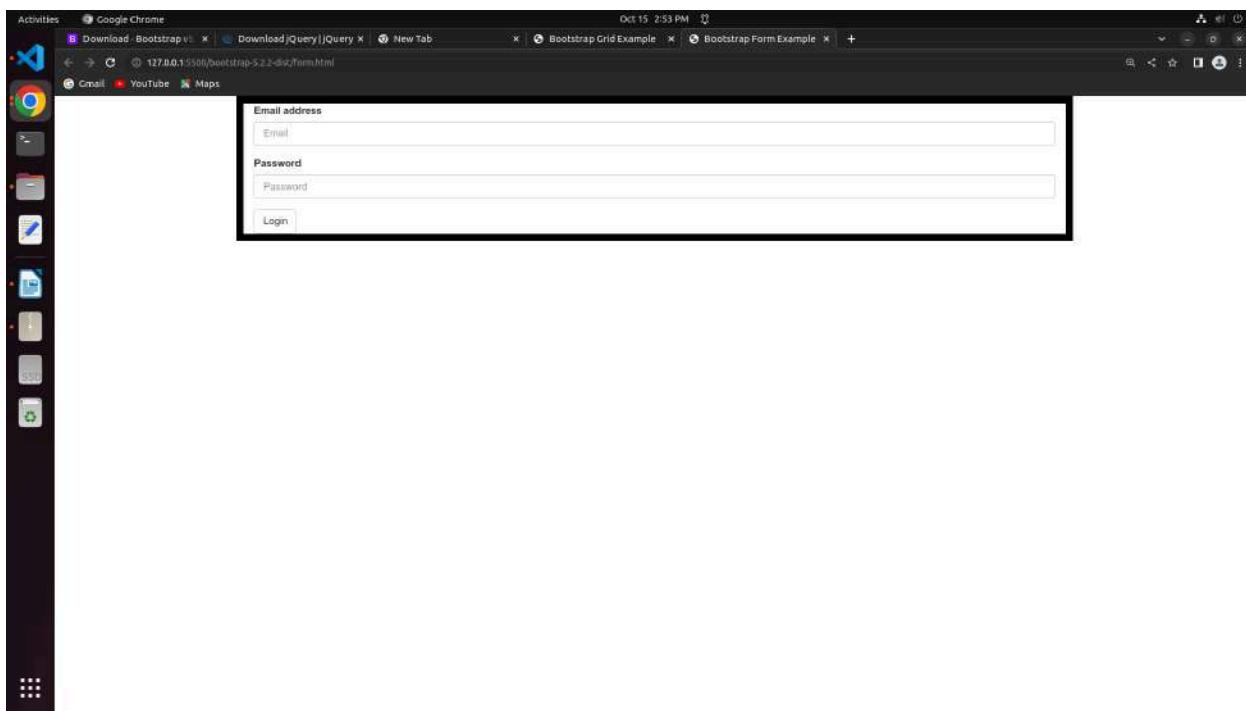
```
.container{  
    border : 10px solid black;  
}
```

**form.html**

```
<html>  
<head>  
    <title>Bootstrap Form Example</title>  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <link rel="stylesheet"  
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">  
    <link rel="stylesheet" href="formStyle.css">  
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>  
    <script  
        src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>  
</head>  
  
<body>  
    <form style="border:25px; margin: 0 auto;">  
        <div class="container">  
            <div class="form-group">  
                <label for="email">Email address</label>  
                <input type="email" class="form-control" placeholder="Email">  
            </div>  
            <div class="form-group">  
                <label for="password">Password</label>  
                <input type="password" class="form-control" placeholder="Password">  
            </div>  
            <button type="submit" class="btn btn-default">Login</button>  
        </div>  
    </form>  
</body>  
</html>
```



## OUTPUT:



### 3. Bootstrap Buttons

#### CODE:

```
<html>
<head>
<title>Bootstrap Button Example</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>
<button class="btn btn-default">default</button>
<button class="btn btn-primary">primary</button>
<button class="btn btn-danger">danger</button>
<button class="btn btn-success">success</button>
<button class="btn btn-info">info</button>
<button class="btn btn-warning">warning</button>
<button class="btn btn-link">Link</button>
```



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

## OUTPUT:



## 4.Bootstrap Navbars

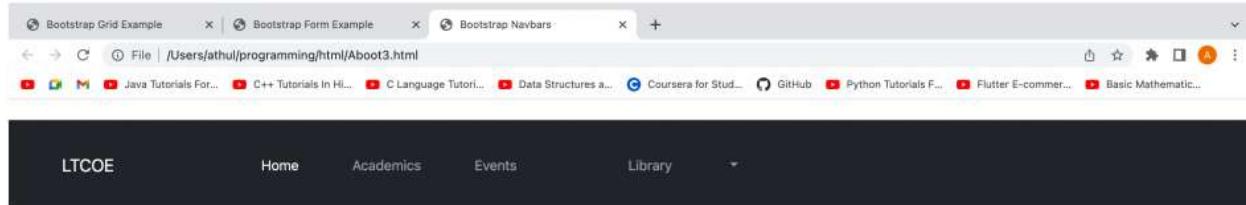
### CODE:

```
<html>  
  <head>  
    <meta name="viewport" content="width=device-width, initial-scale=1">  
    <!-- Bootstrap CSS -->  
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztQTwFspd3yD65VohhpooC0mLASjC" crossorigin="anonymous">  
  <title>Bootstrap Navbars</title>  
  </head>  
  <body>  
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">  
      <div class="container-fluid">  
        <a class="navbar-brand" href="#">LTCOE</a>  
      </div>  
      <div class="collapse navbar-collapse" id="navbarNavDropdown">  
        <ul class="navbar-nav">  
          <li class="nav-item">  
            <a class="nav-link active" aria-current="page" href="#">Home</a>  
          </li>  
          <li class="nav-item">  
            <a class="nav-link" href="#">Academics</a>  
          </li>  
          <li class="nav-item">  
            <a class="nav-link" href="#">Events</a>  
          </li>  
          <li class="nav-item dropdown">  
            <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-expanded="false">
```



```
Library
</a>
<ul class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
<li><a class="dropdown-item" href="#">Online</a></li>
<li><a class="dropdown-item" href="#">Offline</a></li>
<li><a class="dropdown-item" href="#">LibSpace</a></li>
</ul>
</li>
</ul>
</div>
</div>
</nav>
<!-- Bootstrap JavaScript plugin --&gt;
&lt;script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8NI+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+Jc
Xn/
tWtlaxVXM" crossorigin="anonymous"&gt;&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

#### OUTPUT:



## 5.Bootstrap Breadcrumbs

#### CODE:

```
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Bootstrap CSS --&gt;
&lt;link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-EVSTQN3/
azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWspd3yD65VohhpuuC0mLASjC"
crossorigin="anonymous"&gt;
&lt;title&gt;Bootstrap Navbars&lt;/title&gt;
&lt;/head&gt;</pre>
```

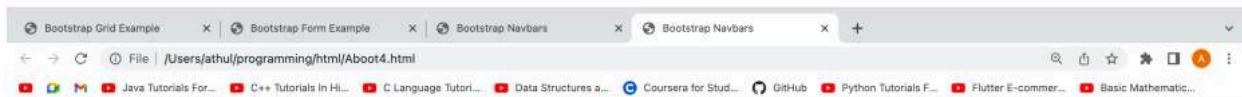


```
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<div class="container-fluid">
<a class="navbar-brand" href="#">LTCOE</a>

<div class="collapse navbar-collapse" id="navbarNavDropdown">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="#">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Academics</a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Events</a>
</li>
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-expanded="false">
Library
</a>
<ul class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
<li><a class="dropdown-item" href="#">Online</a></li>
<li><a class="dropdown-item" href="#">Offline</a></li>
<li><a class="dropdown-item" href="#">LibSpace</a></li>
</ul>
</li>
</ul>
</div>
</div>
</nav>
<!-- Bootstrap JavaScript plugin --&gt;
&lt;script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8NI+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+Jc
Xn/
tWtlaxVXM" crossorigin="anonymous"&gt;&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```



## OUTPUT:



Home

[Home](#) / Library

[Home](#) / [Library](#) / Data



## EXPERIMENT NO.04 - JAVASCRIPT

### 4a-WindowObject

#### 1.windowObject

**CODE:windowObject.html**

```
<html>
<head>
<title>Java Script</title>
</head>
<body onresize="openGoogle()">
<script>
    var iHeight = window.innerHeight;
    var oHeight = window.outerHeight;
    function openGoogle(){
        console.log('inner height : ' + iHeight);
        console.log('Outer height : ' + oHeight);
    }
</script>
</body>
</html>
```

#### **OUTPUT:**

The screenshot shows a desktop environment with a terminal window and a browser window. The terminal window is titled 'Console' and shows the output of the JavaScript code. The browser window is titled 'JavaScript' and displays the source code of 'windowsObject.html'.

**Terminal Output (Console):**

```
Thu 9:09 PM ●
inner height : 668
Outer height : 743
```

**Browser Source (JavaScript):**

```
<html>
<head>
<title>Java Script</title>
</head>
<body onresize="openGoogle()">
<script>
    var iHeight = window.innerHeight;
    var oHeight = window.outerHeight;
    function openGoogle(){
        console.log('inner height : ' + iHeight);
        console.log('Outer height : ' + oHeight);
    }
</script>
</body>
</html>
```



The screenshot shows a terminal window titled "Text Editor" with the file "/home/computer/Documents/CSE-AIML/TE/AIML57\_SUDHAM/WCNL/exp03/windowsObject.html" open. The code in the terminal is as follows:

```
<html>
<head>
<title>Java Script</title>
</head>
<body onresize="openGoogle()">
<script>
    var iHeight = window.innerHeight;
    var oHeight = window.outerHeight;
    function openGoogle(){
        console.log('inner height : ' + iHeight);
        console.log('Outer height : ' + oHeight);
    }
</script>
</body>
</html>
```

The terminal's output window shows repeated log entries for both inner and outer heights, alternating between 668 and 741, indicating the window's dimensions as it is resized.

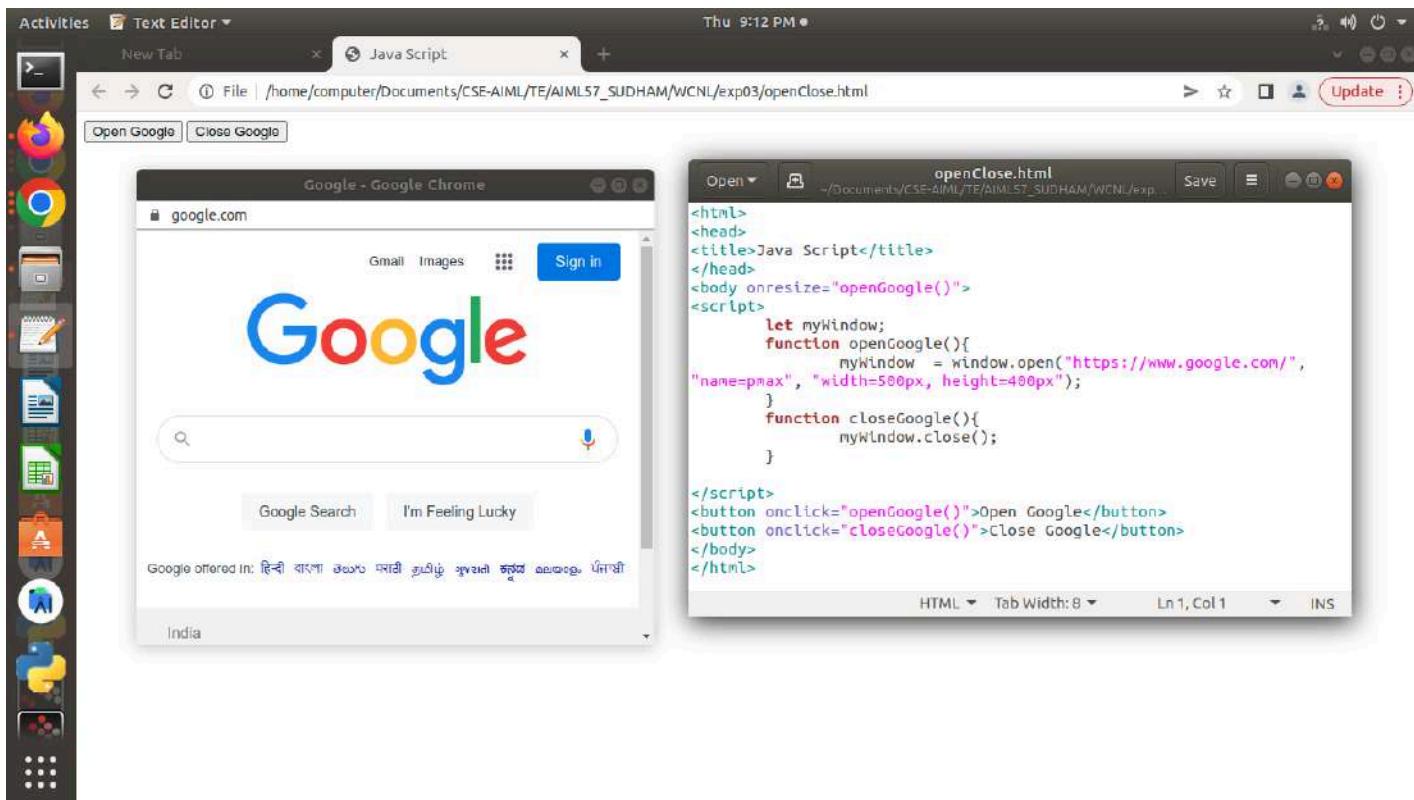
## 2.openCloseGoogle

### CODE:windowObject.html

```
<html>
<head>
<title>Java Script</title>
</head>
<body onresize="openGoogle()">
<script>
    let myWindow;
    function openGoogle(){
        myWindow = window.open("https://www.google.com/", "name=pmax",
        "width=500px, height=400px");
    }
    function closeGoogle(){
        myWindow.close();
    }
</script>
```

```
<button onclick="openGoogle()">Open Google</button>
<button onclick="closeGoogle()">Close Google</button>
</body>
</html>
```

### OUTPUT:



### 3.validation

#### CODE:validation.html

```
<html>
<head>
<title>jvalidation</title>
</head>
<body>
<form action=" " name="myform">
username=<input type="text", name="u_field">
<input type="submit",name="sbtn",value="Submit">
</form>
<script type="text/javascript">
function validatedata()
{
```



```
var ptrn=/^(a-zA-Z)*$/;
if(ptrn.test(document.myform.u_field.value))
document.myform.u_field.style.backgroundColor="green";
else
document.myform.u_field.style.backgroundColor="red";
}
document.myform.u_field.addEventListener("keyup",validatedata);
</script>
</body>
</html>
```

### OUTPUT:

The screenshot shows a Linux desktop environment with a terminal window and a web browser window.

The terminal window (Activities - Text Editor) contains the following validation script:

```
username=09876 Submit
```

```
validation.html
<html>
<head>
<title>jsvalidation</title>
</head>
<body>
<form action=" " name="myform">
username:<input type="text", name="u_field">
<input type="submit",name="sbtn",value="Submit">
</form>
<script type="text/javascript">

function validatedata()
{
var ptrn=/^(a-zA-Z)*$/;
if(ptrn.test(document.myform.u_field.value))

document.myform.u_field.style.backgroundColor="green";

else

document.myform.u_field.style.backgroundColor="red";
}

document.myform.u_field.addEventListener("keyup",validatedata);
</script>
</body>
</html>
```

The browser window (jsvalidation) shows the validation result for the input "09876". The background color of the input field is green, indicating it is valid according to the regular expression.



## **4b-Loops**

### **1.CODE:FOR IN LOOP**

```
<html>
<head>
    <title>For In Loops</title>
    <style>
        html {
            background-color: black;
            color: white;
        }
    </style>
</head>
<body>
    <h2>JavaScript For In Loop</h2>
    <script>
        let student = {
            lname: "Sudham",
            fname: "Singh",
            age: 20
        };
        let txt = "";
        student["email"] = "sudhamsingh2412@gmail.com";
        for (let i in student) {
            txt += i + " : " + student[i] + "<br>";
        }
        document.write(txt);
    </script>
</html>
```

**OUTPUT:**

```
JavaScript For In Loop

lname : Sudham
fname : Singh
age : 20
email : sudhamsingh2412@gmail.com
```



## 2.CODE:FOR LOOP:

```
<!DOCTYPE html>
<head>
    <title>For Loops</title>
    <style>
        body {
            background-color: black;
            color: white;
        }
    </style>
</head>
<body>
    <h2>JavaScript For In Loop</h2>
    <script>
        document.write("Starting Loop" + "<br>");
        for (let count = 1; count <= 10; count+=2) {
            document.write("Current count : " + count);
            document.write("<br>");
        }
        document.write("Loop Stopped!");
    </script>
</body>
</html>
```

## OUTPUT:

### JavaScript For In Loop

```
Starting Loop
Current count : 1
Current count : 3
Current count : 5
Current count : 7
Current count : 9
Loop Stopped!
```

## 3.CODE:FOR UP LOOP

```
<!DOCTYPE html>
<head>
    <title>For Of Loops</title>
    <style>
        body {
```



```
background-color: black;
color: white;
}
</style>
</head>
<body>
<h2>JavaScript For Of Loop</h2>
<p>Print the array elements: </p>
<script>
const nums = [1, 2, 3, 4, 5, 6, 7, 8];
let text = "";
for (let x of nums) {
  text += x + "<br>";
}
document.write(text);
</script>
</body>
</html>
```

**OUTPUT:**

## JavaScript For Of Loop

Print the array elements:

```
1
2
3
4
5
6
7
8
```

**4.CODE:CLASS:**

```
<!DOCTYPE html>
<head>
  <title>Class</title>
  <style>
    body {
      background-color: rgba(0, 0, 0, 0.877);
      color: white;
    }
    h1 {
      color: cyan;
    }
  </style>
</head>
<body>
<h1>Hello World</h1>
</body>
</html>
```



```
        }
    </style>
</head>
<body>
    <h1>JavaScript Statements</h1>
    <h2>The class Statement</h2>
    <script>
        // Super
        class Vehicle {
            constructor(brand) {
                this.carName = brand;
            }
            info() {
                document.write("Super Class <br>")
                document.write(`My Vehicle is: ${this.carName} <br>`);
            }
        }
        class Car extends Vehicle {
            constructor(Brand) {
                super(Brand);
                this.noDoors = 4;
                this.noWheels = 4;
            }
            info() {
                super.info();
                document.write("Sub-class <br>")
                document.write(`No of Doors: ${this.noDoors} <br>`);
                document.write(`No of Wheels: ${this.noWheels} <br>`);
            }
        }
        car1 = new Car("Mini Cooper");
        car1.info();
        vehicle1 = new Vehicle("Mercedes-Benz");
        vehicle1.info();
    </script>
</body>
</html>
```



**OUTPUT:**

# JavaScript Statements

## The class Statement

Super Class

My Vehicle is: Mini Cooper

Sub-class

No of Doors: 4

No of Wheels: 4

Super Class

My Vehicle is: Mercedes-Benz

### 5.CODE:IF-ELSE:

```
<!DOCTYPE html>
<head>
  <title>If-Else</title>
  <style>
    body {
      background-color: black;
      color: white;
    }
  </style>
</head>
<body>
  <h2>If-Else</h2>
  <script>
    let age = 23;
    if (age >= 18) {
      document.write("You are eligible to vote");
    } else {
      document.write("You are not eligible to vote");
    }
  </script>
</body>
</html>
```

**OUTPUT:**

If-Else

You are eligible to vote



## 4c-Generators

### 1.CODE:generators.html

```
<html>
<body>
<script>
function* test(){
    yield "yield no.1";
    yield "yield no.2";
    yield "yield no.3";
}
let generator = test();
for (var i = 0; i < 3; i++){
    let g = generator.next();
    console.log(g.value);
    console.log(g);
}
</script>
</body>
</html>
```

### OUTPUT:

The screenshot shows the Google Chrome DevTools Console tab. The output of the generator code is displayed as follows:

```
yield no.1
  ↘ {value: 'yield no.1', done: false} i
    done: false
    value: "yield no.1"
  ► [[Prototype]]: Object
yield no.2
  ↘ {value: 'yield no.2', done: false} i
    done: false
    value: "yield no.2"
  ► [[Prototype]]: Object
yield no.3
  ↘ {value: 'yield no.3', done: false} i
    done: false
    value: "yield no.3"
  ► [[Prototype]]: Object
```

The output shows three iterations of the generator function, each returning an object with a 'value' property containing the string 'yield no.1', 'yield no.2', or 'yield no.3' respectively, and a 'done' property set to 'false'. The 'Console' tab is selected in the DevTools header.



## 2.CODE:whileLoopGenerators

```
<!DOCTYPE html>
<head>
<title>
    Generators
</title>
</head>
<body>
    <h1>Unlimited yields</h1>
    <script>
        function* generateit(){
            let nextnum=300;
            while(true){
                yield(nextnum++);
            }
        }
        let g = generateit()
        console.log(g.next());
        console.log(g.next());
        console.log(g.next());
    </script>
</body>
</html>
```

### OUTPUT:

The screenshot shows the Google Chrome Developer Tools open in a new tab. The 'Console' tab is selected. Three generator objects are listed, each with properties `done` and `value`. The first object has `value: 300`, the second has `value: 301`, and the third has `value: 302`. The objects are shown as expandable structures with their prototypes.

Object	Value	File
Object 1	done: false value: 300	generators2.html:17
Object 2	done: false value: 301	generators2.html:18
Object 3	done: false value: 302	generators2.html:19



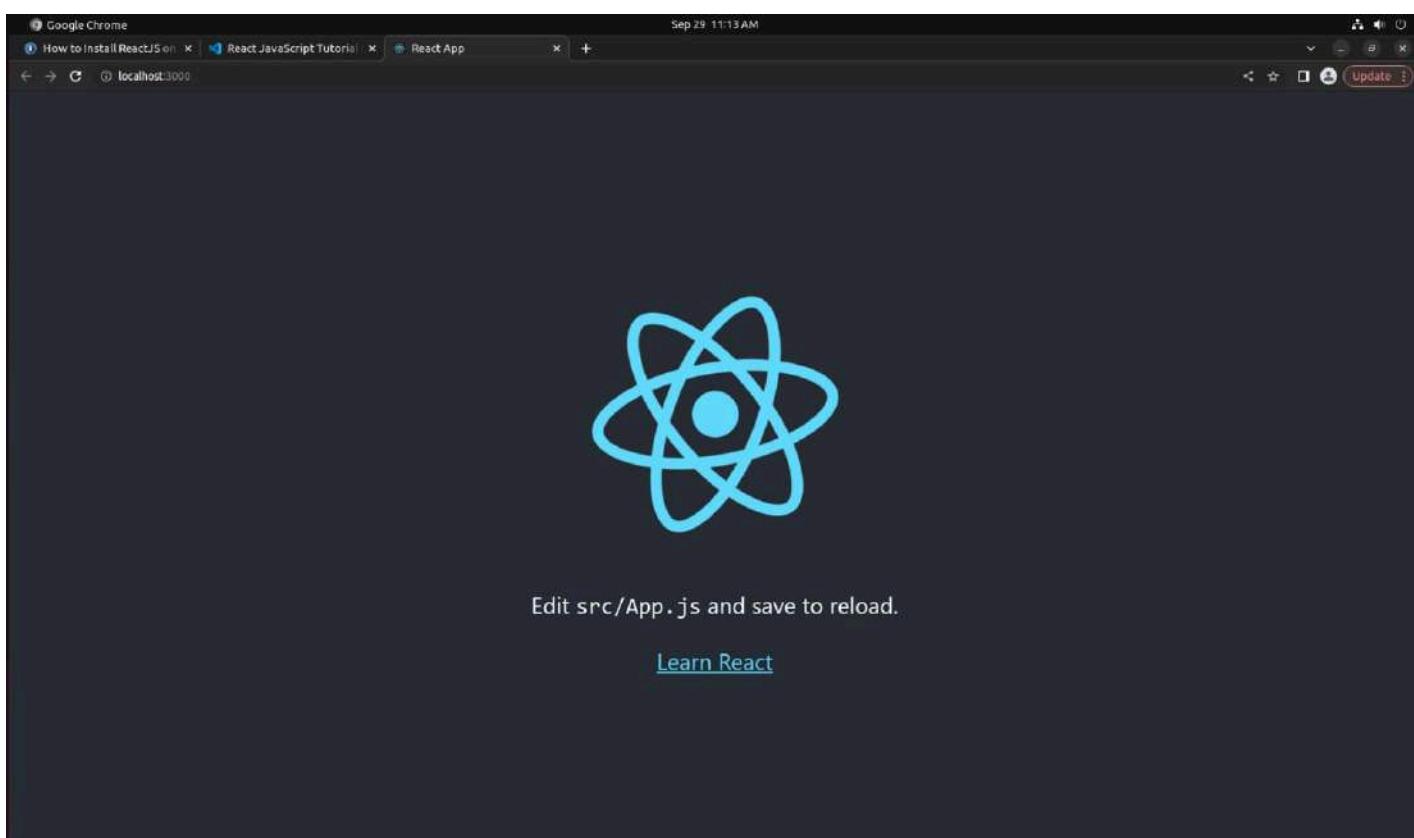
## EXPERIMENT NO.05 - REACT.JS

### 1. Installation

#### CODE:

```
sudo apt update  
sudo apt install nodejs // [node –version]  
sudo apt install npm // [npm –version]  
sudo npm -g install create-react-app  
create folder project  
cd project  
create-react-app project_name // small letter  
cd project_name  
npm start
```

#### OUTPUT:





Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER JS App.js (1) launch.json index.js

```
src
  - index.js
    - App.js
    - App.css
    - App.test.js
    - index.css
    - logo.svg
    - reportWebVitals.js
    - setupTests.js
    - gitignore
  - package-lock.json
  - package.json
  - README.md
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <h1>
    Hello, Sudham Welcome to the React World!
  </h1>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

PROBLEMS DEBUG CONSOLE TERMINAL COMMENTS

[eslint] src/index.js Line 4:8: 'App' is defined but never used [no-unused-vars](#)  
Search for the [keywords](#) to learn more about each warning.  
To ignore, add [//eslint-disable-next-line](#) to the line before.  
WARNING in [eslint] src/index.js Line 4:8: 'App' is defined but never used [no-unused-vars](#)  
webpack compiled with 1 warning

OUTLINE TIMELINE

Launch Chrome against localhost (PROJECT) Taken at 9:05 PM

Google Chrome

New Tab React App +

localhost:3000

Gmail YouTube Maps

Hello, Sudham Welcome to the React World!

## 2.ReactDom(components)

**CODE:**

### NoPage.js

```
const NoPage = () => {
  return <h1>404</h1>;
};
```

export default NoPage;

### About.js

```
const About = (props = "") => {
  return (
    <h1>About {props.name}</h1>
  );
};
```

export default About;

### Blogs.js

```
const Blogs = () => {
  return <h1>Blogs</h1>
```



```
}
```

```
export default Blogs;
```

**Contact.js**

```
const Contact = () => {
  return <h1>Contact</h1>;
};
```

```
export default Contact;
```

**Home.js**

```
const Home = () => {
  return <h1>Home</h1>;
};
```

```
export default Home;
```

localhost:3000/NoPage

- [Home](#)
- [Blogs](#)
- [Contact](#)
- [NoPage](#)
- [About](#)

# 404

localhost:3000/About

- [Home](#)
- [Blogs](#)
- [Contact](#)
- [NoPage](#)
- [About](#)

# About

localhost:3000/blogs

- [Home](#)
- [Blogs](#)
- [Contact](#)
- [NoPage](#)
- [About](#)

# Blogs

localhost:3000/contact

- [Home](#)
- [Blogs](#)
- [Contact](#)
- [NoPage](#)
- [About](#)

# Contact

localhost:3000

- [Home](#)
- [Blogs](#)
- [Contact](#)
- [NoPage](#)
- [About](#)

# Home



## EXPERIMENT NO.06 - NODE.JS

### **1.Installation**

To start new project : npminit

To run : npm run start

To print simple hello world!!

#### **CODE:helloworld.js**

```
console.log("hello world!!");
```

#### **OUTPUT:**

```
hello world!!
```

### **2.EventLoop**

#### **CODE:Event\_loop.js**

```
console.log("This is the first statement");
setTimeout(function(){
    console.log("This is the second statement");
}, 1000);
console.log("This is the third statement");
```

#### **OUTPUT:**

```
PS D:\StudyTime\TE\WC\nodejs> node
"d:\StudyTime\TE\WC\nodejs\EventLoop\Event_loop.js"
This is the first statement
This is the third statement
This is the second statement
```

### **CALLBACKS:-**

### **3.Asynchronous**

#### **CODE:**

#### **inputfile1.txt**

Hello!!!

Code for reading a file asynchronously (non-blocking code) in Node.js.

#### **async.js**

```
var fs = require("fs");
fs.readFile('inputfile1.txt', function (err, filedata) {
    if (err) return console.error(err);
    console.log(filedata.toString());
});
console.log("End of Program execution");
```



## OUTPUT :

```
PS D:\StudyTime\TE\WC\nodejs> node "d:\StudyTime\TE\WC\nodejs\async.js"
```

End of Program execution

Hello!!!

Code for reading a file asynchronously (non-blocking code) in Node.js.

## 4.Synchronous

### OUTPUT :

```
node > src > JS assyncjs > ...
1  var fs = require("fs");
2
3  fs.readFile('file1.txt', function (err, filedata) {
4      if (err) return console.error(err);
5      console.log(filedata.toString());
6  });
7  console.log("End of Program execution");
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

```
PS C:\VS Code\WC\node\src> node sync.js
```

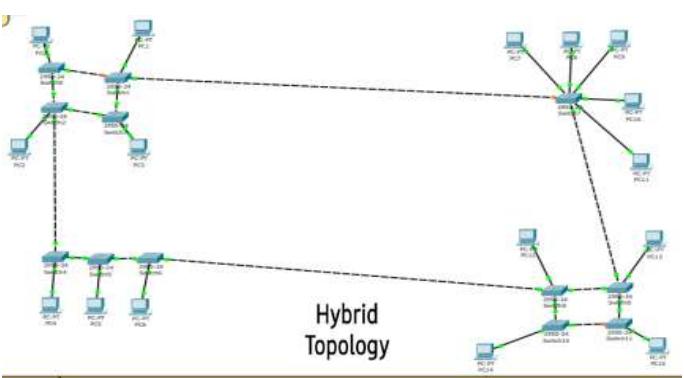
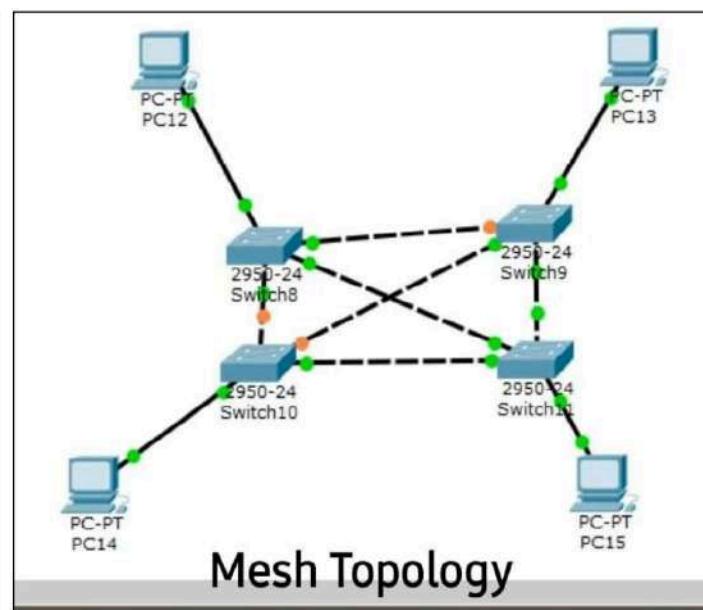
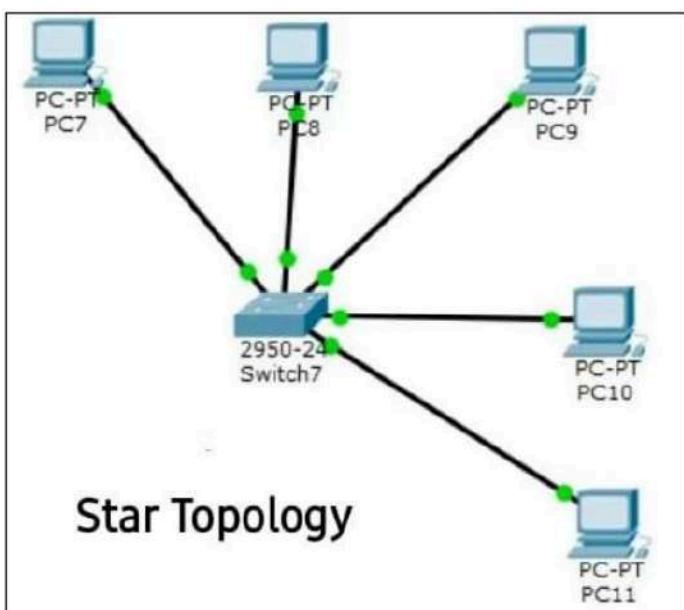
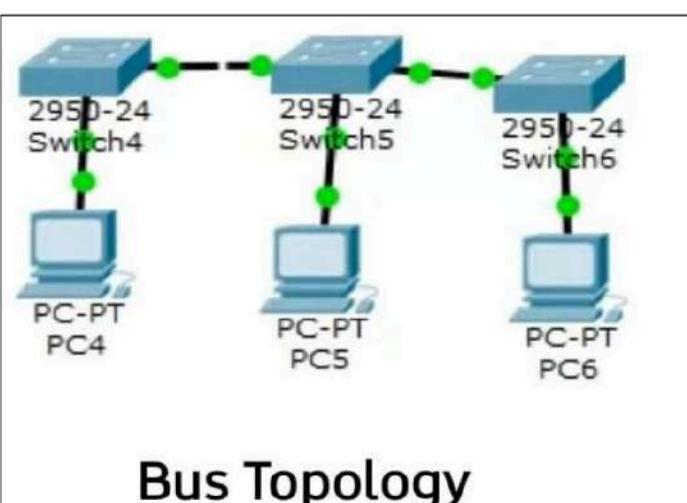
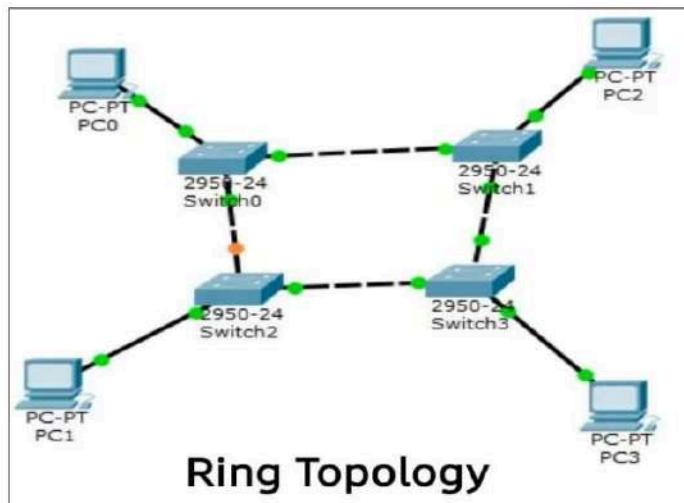
Hello!!!

Code for reading a file synchronously (blocking code) in Node.js.

End of Program execution

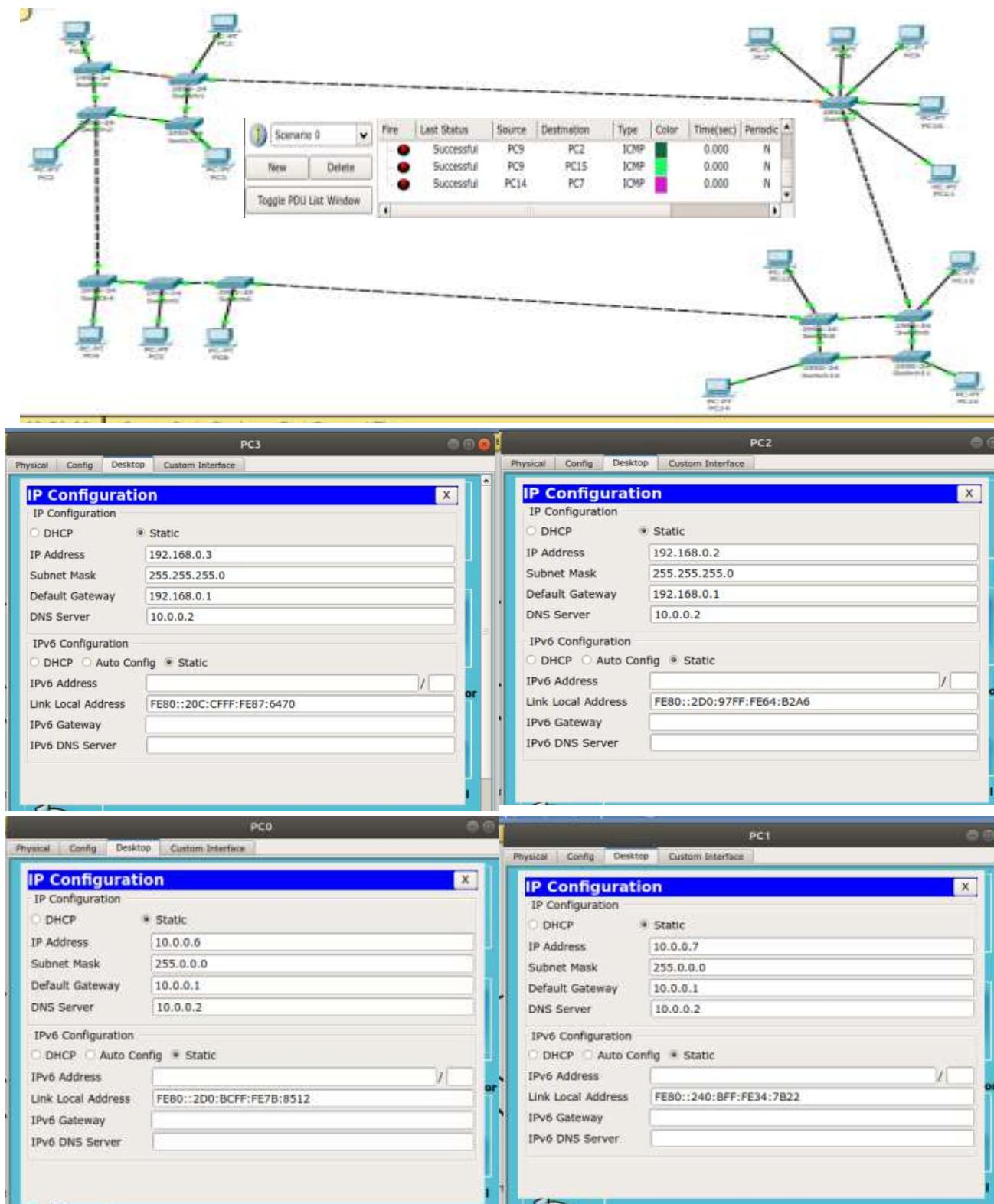
## EXPERIMENT NO.07

**OUTPUT:  
TOPOLOGIES:**



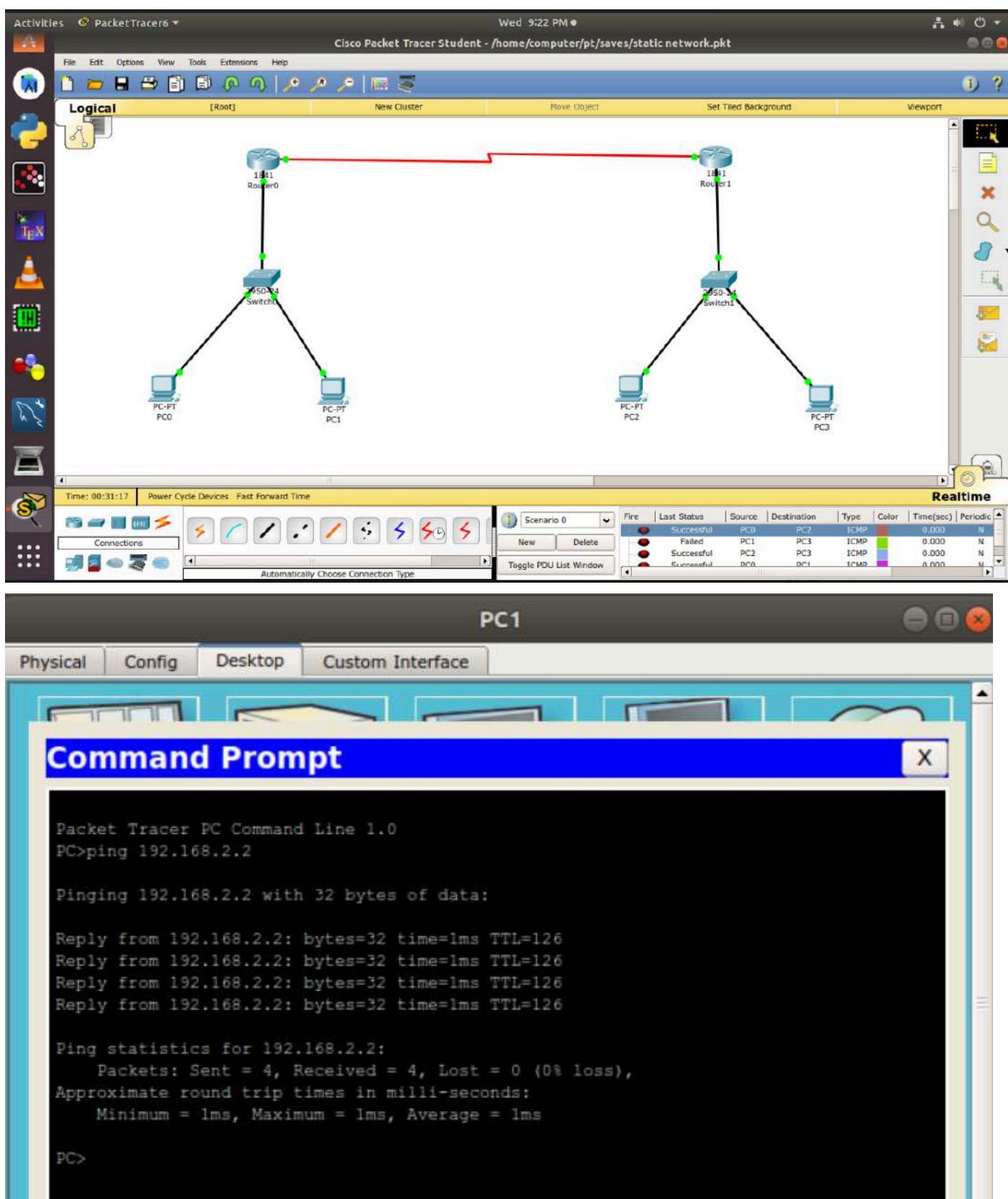


## IP setup and its configuration to a static routing :



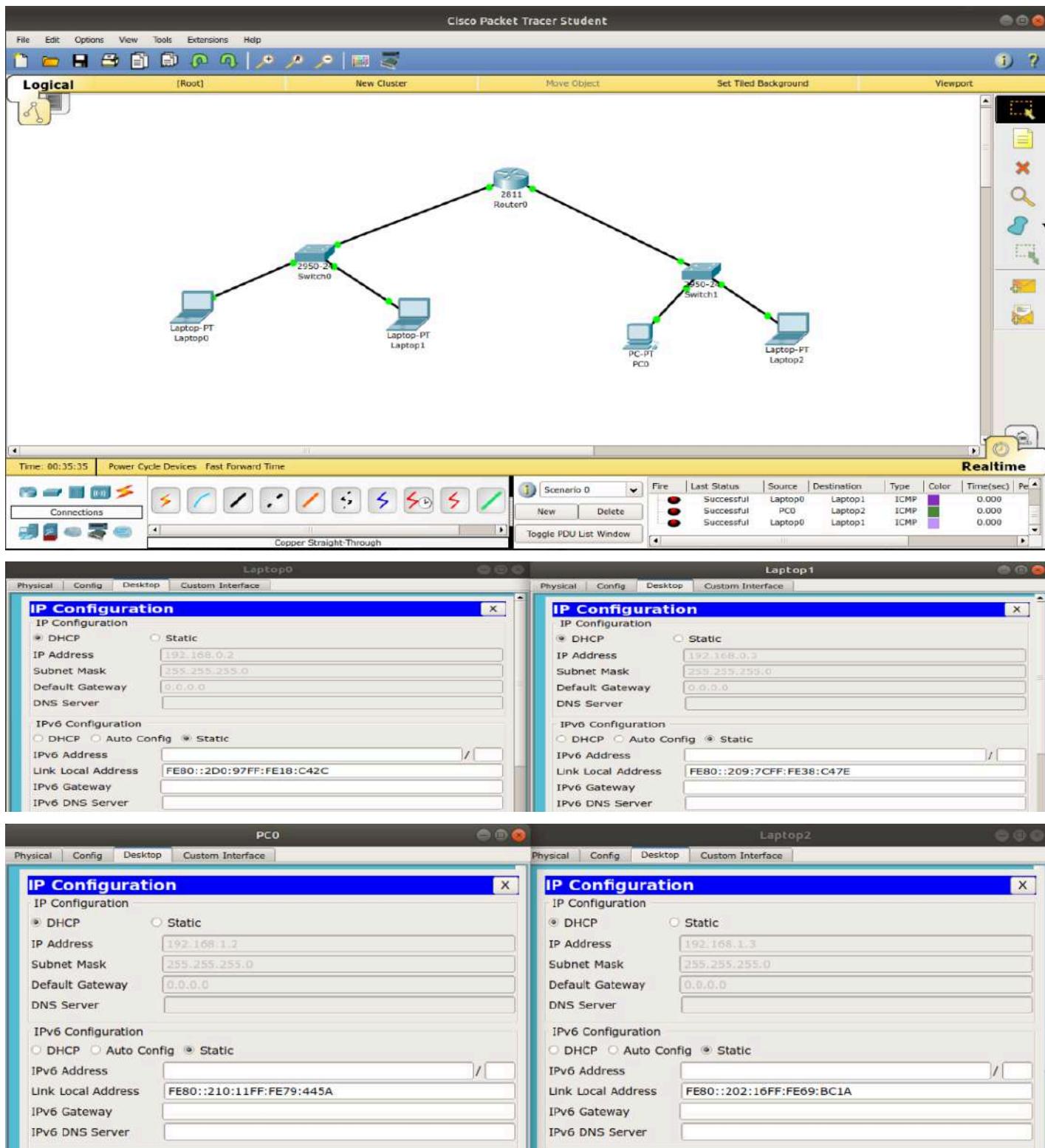


## STATIC ROUTING:



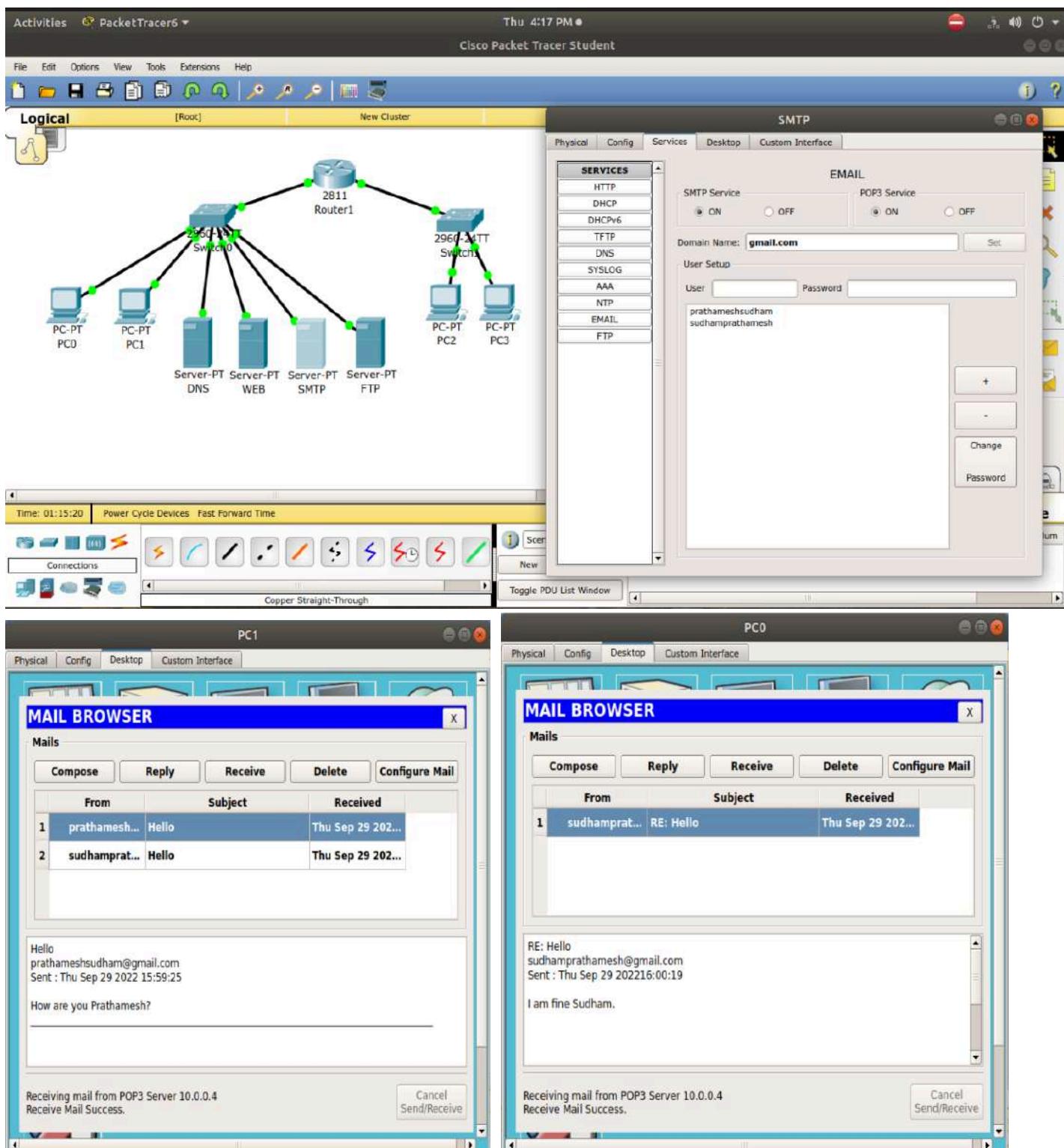
## EXPERIMENT NO.08

### **OUTPUT: CONFIGURATION OF DHCP SERVER**



## EXPERIMENT NO.09

### **OUTPUT: SMTP SERVER-**



## FTP SERVER-

Activities Cisco Packet Tracer Student

Thu 4:17 PM •

**Logical** [Root] New Cluster

File Edit Options View Tools Extensions Help

Router1 (2811) connected to SWC10 (2960) and SWC11 (2960 MTT).

SWC10 connected to PC-PT PC0, PC-PT PC1, Server-PT DNS, Server-PT WEB, Server-PT SMTP, and Server-PT FTP.

SWC11 connected to PC-PT PC2 and PC-PT PC3.

**FTP**

Physical Config Services Desktop Custom Interface

**SERVICES**

- HTTP
- DHCP
- DHCPv6
- TFTP
- DNS
- SYSLOG
- AAA
- NTP
- EMAIL
- FTP

**Service** On

User Setup

Username	Password	Permission
cisco	cisco	RWDNL
prathameshsudham	prathamesh123	RNL

**File**

- 1 csa842-k8.bin
- 2 c1841-advpipervicesk9-mz.124-15.T1.bin
- 3 c1841-ipbase-mz.123-14.T7.bin
- 4 c1841-ipbasek9-mz.124-12.bin

**PC0**

Physical Config Desktop Custom Interface

**Command Prompt**

```

PC>ping 10.0.0.5
Pinging 10.0.0.5 with 32 bytes of data:
Reply from 10.0.0.5: bytes=32 time=1ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128
Reply from 10.0.0.5: bytes=32 time=0ms TTL=128

Ping statistics for 10.0.0.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

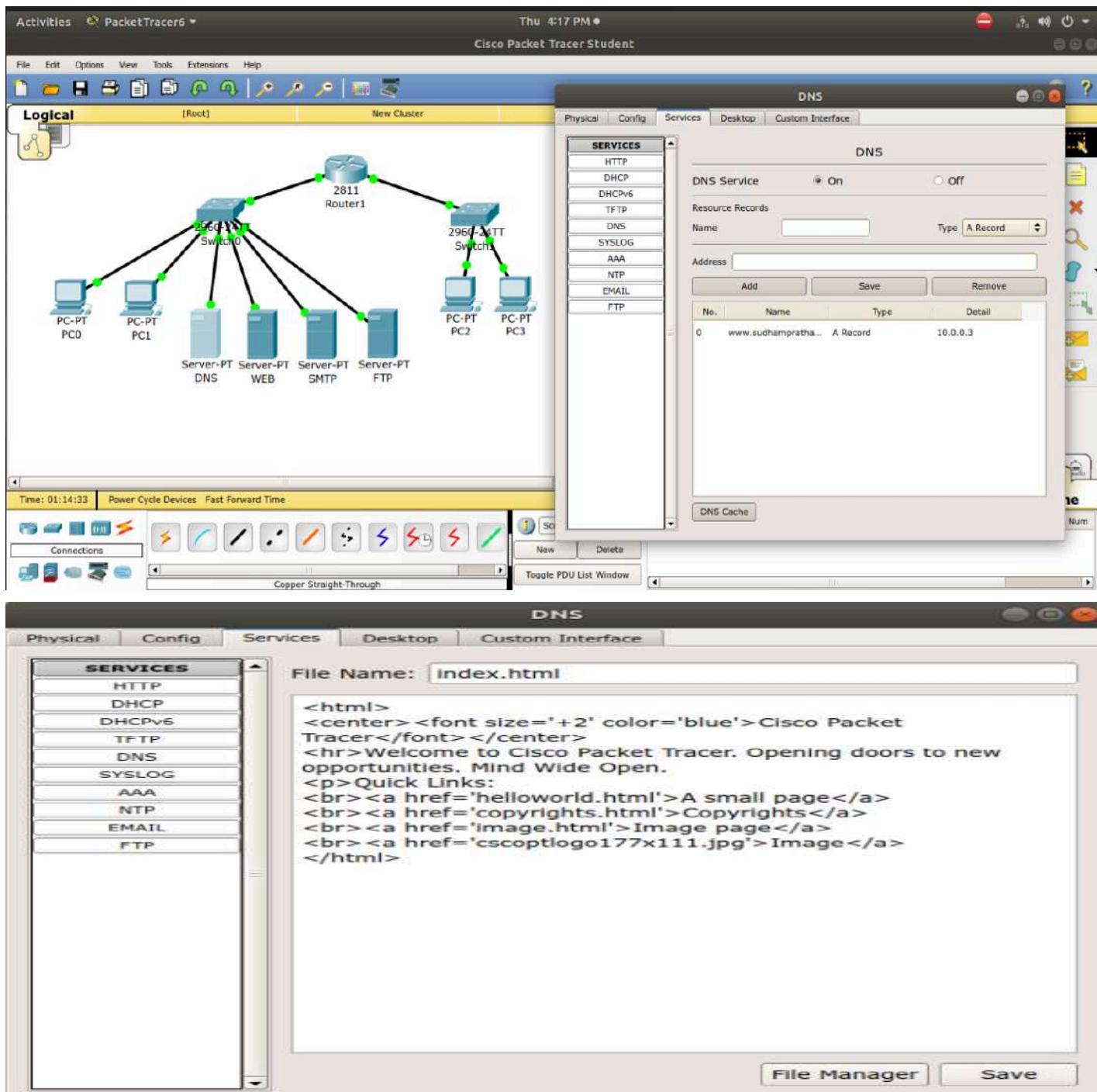
PC>ftp 10.0.0.5
Trying to connect...10.0.0.5
Connected to 10.0.0.5
220- Welcome to PT Ftp server
Username:prathameshsudham
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>dir

Listing /ftp directory from 10.0.0.5:
0 : asa842-k8.bin                                5571584
1 : c1841-advpipervicesk9-mz.124-15.T1.bin      33591768
2 : c1841-ipbase-mz.123-14.T7.bin                13832032
3 : c1841-ipbasek9-mz.124-12.bin                 16599160
4 : c2600-advpipervicesk9-mz.124-15.T1.bin      33591768
5 : c2600-i-mz.122-28.bin                         5571584
6 : c2600-ipbasek9-mz.124-8.bin                  13169700
7 : c2800nm-advpipervicesk9-mz.124-15.T1.bin     50938004
8 : c2800nm-advpipervicesk9-mz.151-4.M4.bin      33591768

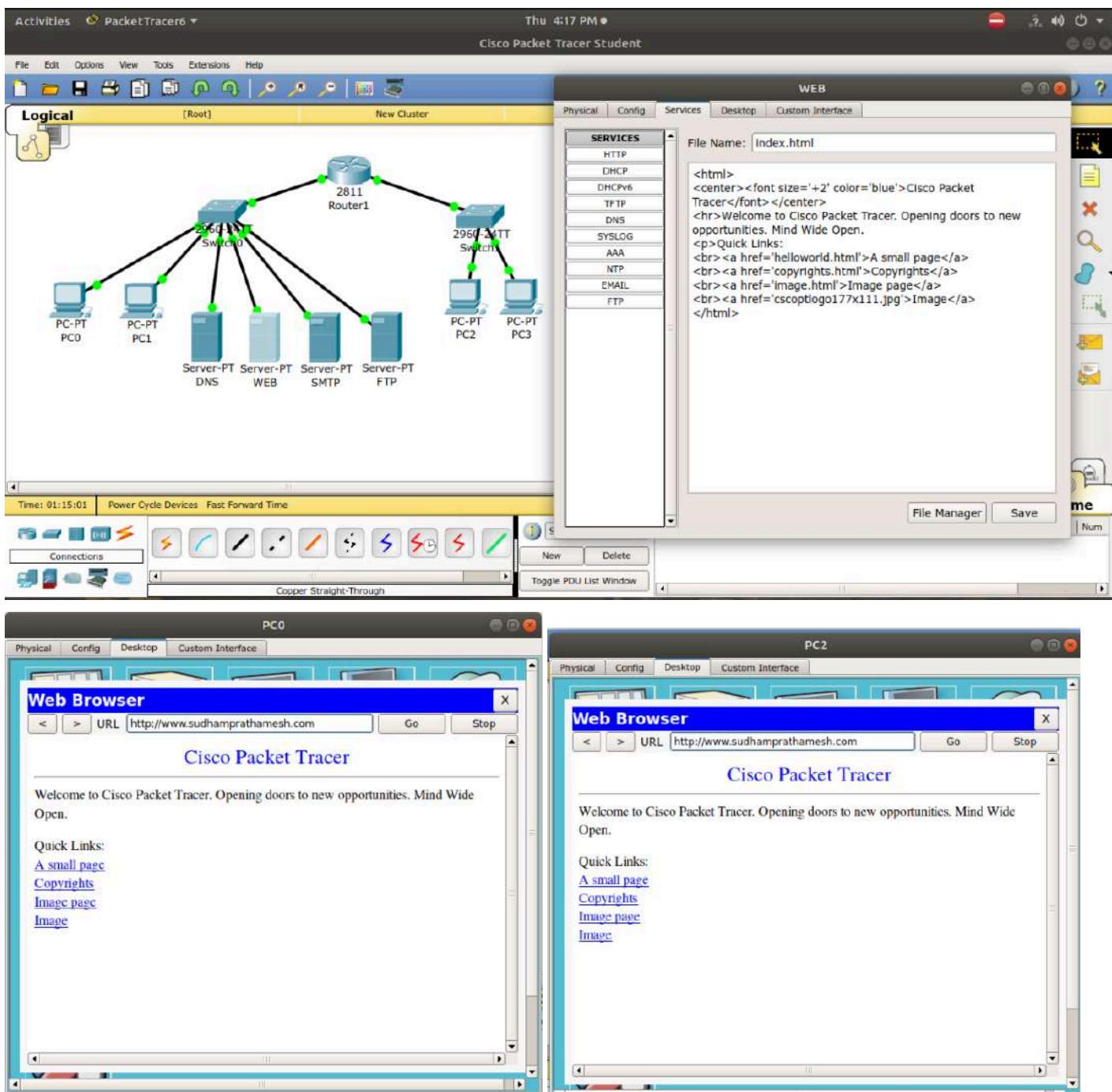
```

## EXPERIMENT NO.10

### OUTPUT: DNS SERVER-



## WEB SERVER-





Name :- Singh Sudham Dharmentra

Branch :- CSE(AIML)

Roll no:- AIML57

Subject :- Web computing [Computer Network]

Topic :- Assignment No. ②

Date of Submission :- 10/22

Signature :- Singh

Q.1] Difference between TCP & UDP.

TCP	UDP
① TCP stands for Transmission Control Protocol	① UDP stands for User Datagram Protocol
② TCP is a connection-oriented protocol	② UDP is Connectionless protocol
③ TCP is highly reliable	③ UDP is <sup>suitable</sup> efficient for transmission
④ TCP is used by HTTP, HTTPS, SMTP, Telnet	④ UDP is used by DNS, DHCP, SNMP, VOIP
⑤ Ordering of packet is maintained by TCP	⑤ UDP has packets independent of each other.
⑥ Speed is slower than UDP	⑥ Speed is faster than TCP
⑦ Header size is of 20 bytes	⑦ Header size is of 8 bytes
⑧ It has Acknowledgment Segments	⑧ It has no Acknowledgment

Q.2] Explain CSMA/CD in detail & also mention its use.

- ① CSMA/CD stands for Carrier Sense Multiple Access / collision Detection, with collision detection being an extension of the CSMA protocol.
- ② This creates a procedure that regulates how communication must take place in a network with a shared transmission medium.
- ③ The extension also regulates how to proceed if collision occurs i.e., when two or more nodes try to send data packets via the transmission medium (bus) simultaneously and they interfere with one another.
- ④ To understand CSMA/CD works, we break down the components:-
- (i) Carrier Sense (CS): The carrier state detection make sure that all network participants check whether the medium is current free; only then does the protocol initiates data transmission.

(ii) Multiple access (MA): - Several participants [computers connected to the network] share a transmission medium.

(iii) Collision detection (CD): - The collision detection is an extension of the original protocol and regulates how to proceed in case data packets happen to collide.

Uses of CSMA/CD:

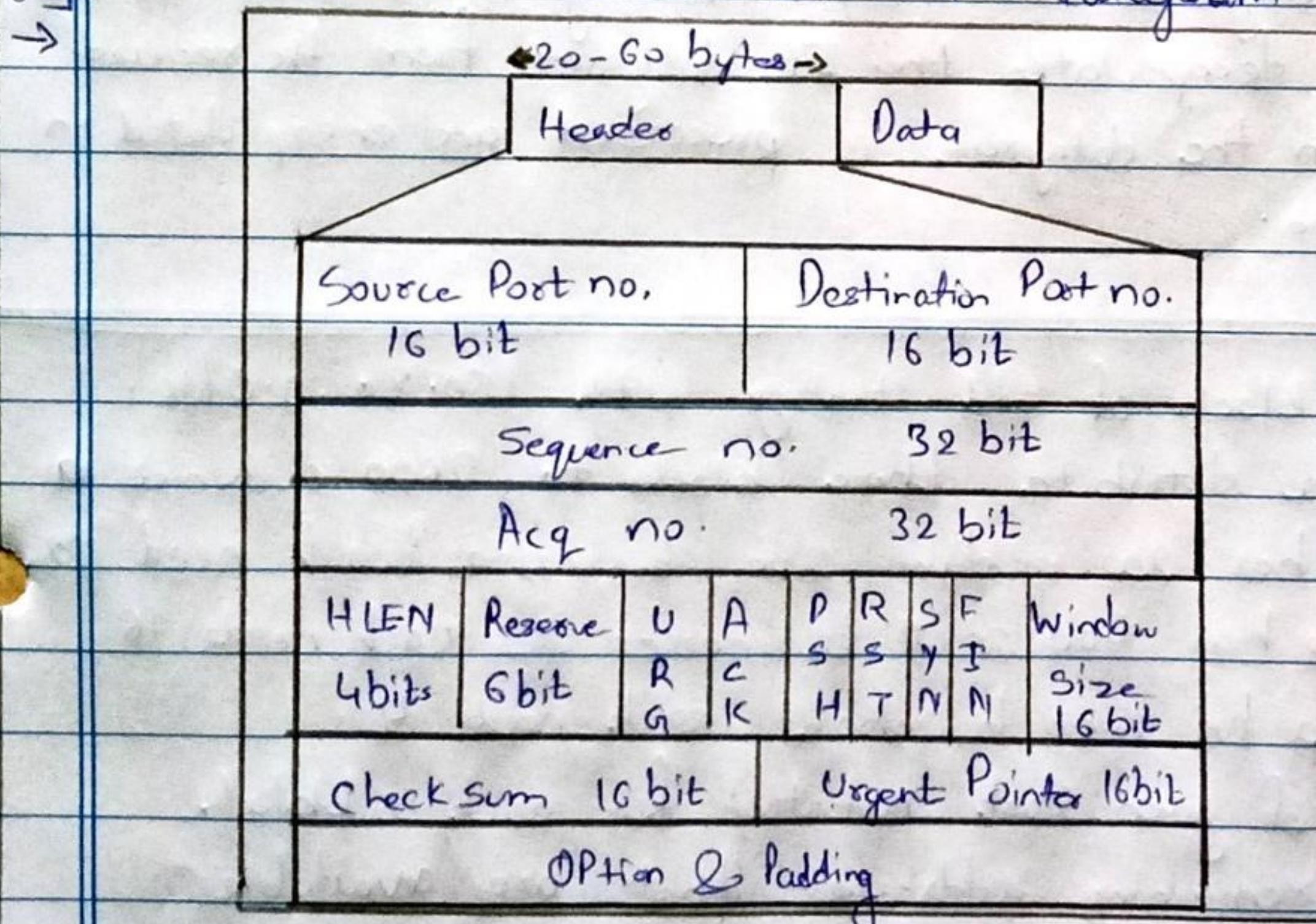
(i) CSMA/CD is technique used for shared medium access.

(ii) CSMA/CD is used in Ethernet.

(iii) It detect collision within a short time.

(iv) It avoids wasteful transmission.

Q.3 Draw header format of TCP datagram.



Q.4 Explain different Network layer functions.

→ Duties of network layer

internetworking    Addressing    Routing    Packetizing    Fragmenting

(i) Internetworking: This is the main duty of network layer. It provides the logical connection between different types of networks.

(ii) Addressing :- Addressing is necessary to identify each device on the internet uniquely. This is similar to a telephone system. The addresses used in the network layer should uniquely & universally define the connection of a computer.

(iii) Routing :- It is a network, there are multiple routes available from a source to destination and one of them is to be chosen. The network layer decides the route to be taken. This is called as routing and it depends on various criterions.

(iv) Packetizing :- As discussed earlier, the network layer encapsulates the packets received from upper layer protocol and makes new packets. This is called packetizing.

(v) Fragmenting : The datagram can travel through different networks. Each router decapsulates the IP datagram from the received frame. Then the datagram is processed and encapsulated in another frame.

Q.5] As ISP granted a block of add starting with 150.80.0.0/16.

The ISP wants to distribute these blocks to 2600 customers as follows:

(a) 1<sup>st</sup> group has 200 medium size businesses & each need 128.

(b) The 2<sup>nd</sup> group has 400 small businesses & each needs 16.

(c) The 3<sup>rd</sup> group has 2000 households each needs 4.

Design the subblock of given slash notation for each subblock and find out the remaining addresses that are available.

$$\rightarrow \text{Total address} = 2^{32-n} = 2^{32-16} = 2^{16} = 65536$$

$$\text{Group } ① : 128 = 2^{7 \rightarrow 9} = 32 - 7 = 25 = n, \text{ subnet.}$$

Business ① : 150.80.0.0/25 .... starting  
150.80.0.127/25 .... ending

Business ② : 150.80.0.128/25  
150.80.0.255/25

Business (200) :-

150.80.99.128/25

150.80.99.255/25

$$\text{Total address} = 200 \times 128 = 25600$$

Group 2 :-  $16 = 2^4 \rightarrow 9$   $32 - 4 = 28 \rightarrow n_2, \text{ subnet}$

Business ① :- 150.80.100.0/28

150.80.100.15/28

Business ② :- 150.80.100.16/28

150.80.100.31/28

:

Business (400) :- 150.80.124.240/28

150.80.124.255/28

$$\text{Total addresses} = 400 \times 16 = 6400$$

Group 3 :-  $4 = 2^2 \rightarrow 9$   $32 - 2 = 30, n_3, \text{ subnet}$

Business ① :- 150.80.125.0/30

150.80.125.3/30

:

Business (64) :- 150.80.125.252/30

150.80.125.255/30

:

Business (192) :- 150.80.155.252/30

150.80.155.255/30

:

Business (2000) :- 150.80.156.60/30

150.80.156.63/30

$$\text{Total address} = 2000 \times 4 = 8000$$

$$\text{Total remaining address} = 65536 - 40,000$$

= 25536

Q.6] Explain IPV4 head format to Draw.  
→ 20-60 bytes

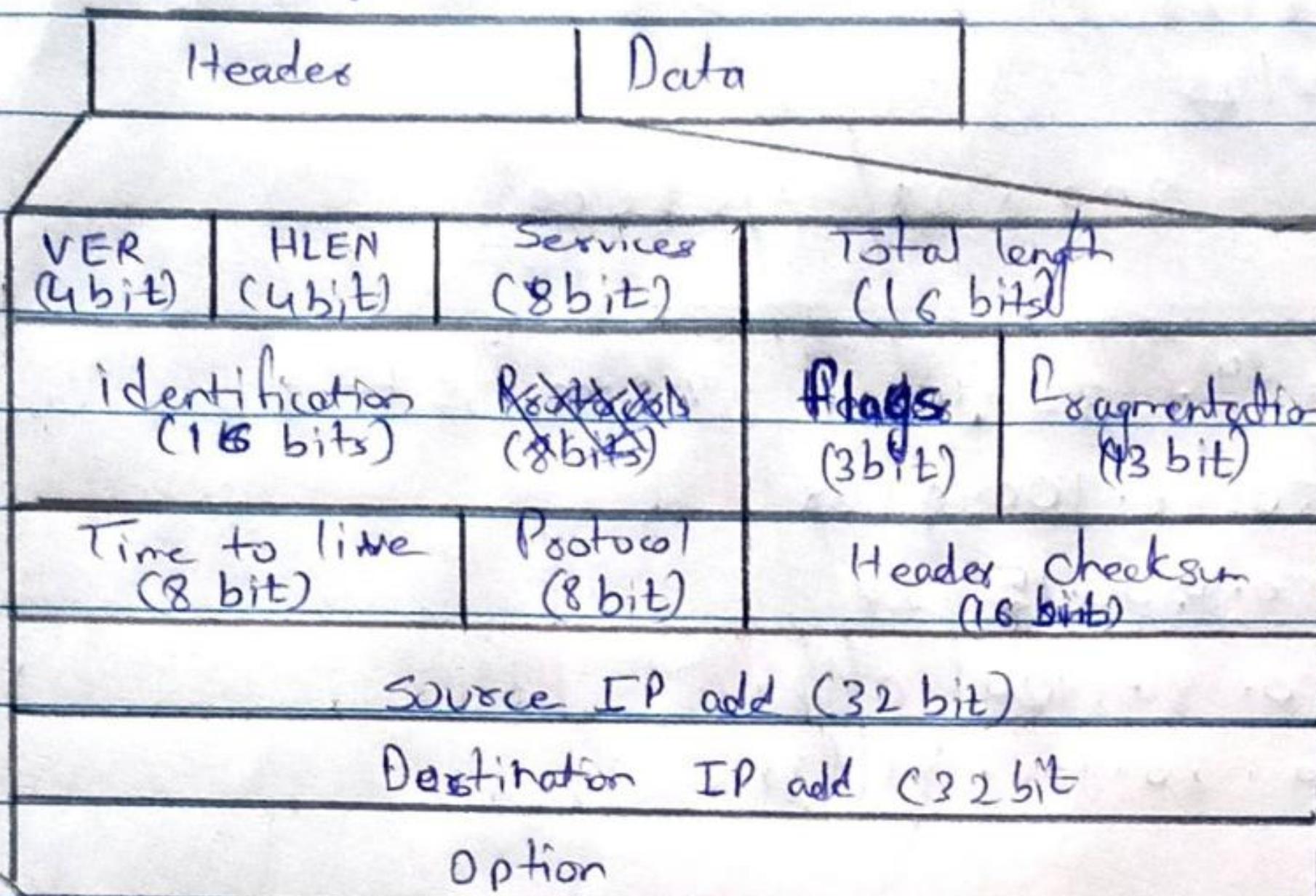


Fig. IPV4 header format.

- ① VER (4 bits): This field defines the version of IP (current version).
- ② HLEN (4bit) : Header length - This field defines the length of datagram headers in 4-byte word
- ③ Differential Services (8 bit) : This field defines the class of datagram for quality of services.
- ④ Total length (16 bit) : This field defines the total length of IP datagram
- ⑤ Identification (16 bit): This field identifies the datagram originating from the source host.
- ⑥ Flag (3 bit): Flag should be 1st bit reserved & and it should 0.
- ⑦ Fragmentation (4bit): It shows the relative position of the fragment with respect to whole datagram.
- ⑧ Time to live (8 bit): It is used to avoid looping network
- ⑨ Protocol (8 bit) :- Tells the network layer at the destination host, to which protocol this packet belongs.
- ⑩ Header checksum (16 bits): It is used to keep checksum value of entire header.
- ⑪ Source address (32 bit):- A 32-bit add of the sender of the packet
- ⑫ Destination (32 bit) :- A 32 bit add of the receiver of the packet.
- ⑬ Option :- This option field is contain for security, Record route, Time stamp, etc.

(Q7) Write a short note on NAT

- ① NAT (Network Address Translation) was designed as a short term solution for this issue by providing a mechanism of connecting more than one computer to an IP network with the help of either a single unique public IP address, or a small number of uniquely public IP addresses.
- ② NAT is generally used in different types of corporations, where a NAT router is placed at the edge of the private network and transforms the IP addresses which are attached to packets.
- ③ The NAT router, which performs as an effective agent between the internet and the local network.
- ④ For eg, when a computer is placed in a private network it request data from public network.
- ⑤ When the information is been requested is returned, the NAT router checks its internal list of public to private address mappings for the purpose of forwarding the information to the suitable computer.

Type of NAT - (i) static NAT environments,

(ii) Dynamic NAT environments.

(i) Static NAT environments :-

In static NAT environments, the NAT router maps private and public addresses on a one-to-one basis, that is, the private address of a given device always maps to some public address.

(ii) Dynamic NAT environments :-

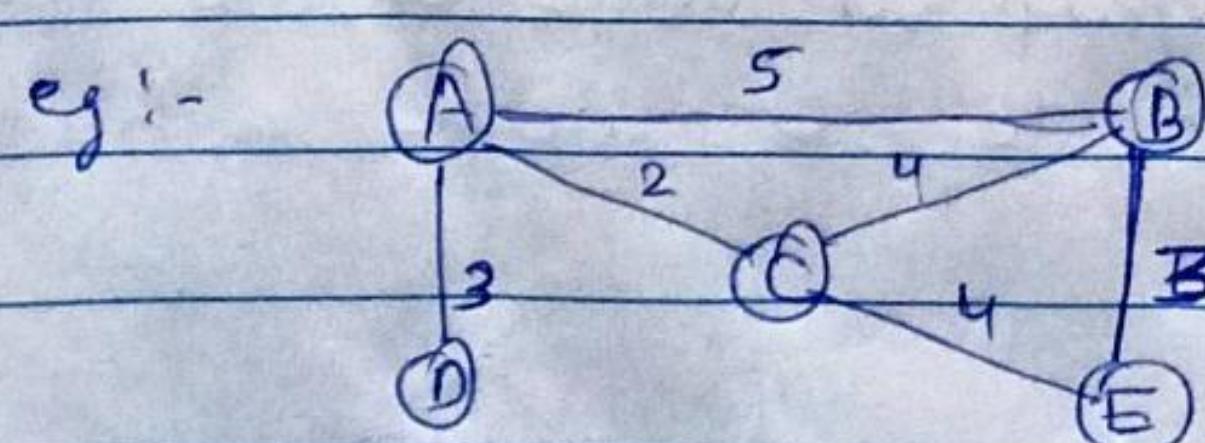
In a dynamic NAT environments, the NAT router dynamically allocate public IP addresses, from a group of addresses, to devices on the private network that wish to communicate with the public network.

Q.8] Difference between IPv4 & IPv6.

→	IPv4	IPv6
①	IPv4 has 32 bit add length	① IPv6 has 128 bit address length
②	It has support for manual & DHCP configuration	② It has support for auto-configuration & remembering.
③	Availability of max 4.3 billion addresses	③ It has limitless number of IP address available.
④	Security is dependent on application.	④ Security purpose IPSEC is built in IPv6 protocol.
⑤	Optional fields are available.	⑤ Optional field are not available.
⑥	Broadcasting message transmission	⑥ Multicasting message transmission
⑦	Address represent in decimal form.	⑦ Address represent in hexadecimal form.

Q.9] Short note on Dijkstra Algorithm.

- ① Dijkstra Algorithm is also known as shortest path.
- ② Each node has to be labelled with its distance. The distance must be non-negative. Initially, no paths are known, so all nodes are labelled with infinity.
- ③ As the algorithm proceeds and path are found, the labels may change, reflecting better paths. A label may be either tentative or permanent. Initially, all labels are tentative.
- ④ When it discovered that label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter, network can have more than one shortest path.



To find shortest path from  
A to E

Permanent list	tentative list
-	A(0)
A(0)	B(5), C(2), D(3)
A(0), C(2)	B(5), D(3), E(6)
A(0), C(2), D(3)	B(5), E(6)
A(0), C(2), D(3), B(5)	E(6)
A(0), C(2), D(3), B(5), E(6)	-

Q.10] Explain Cisco Service Oriented Network Architecture.

- ① The Cisco Services-oriented Network Architecture (SONA) approach provides a standard paradigm for designing networks that link network services with applications to drive business value by:
  - (i) Establishing rapid adoption & deployment of new application services.
  - (ii) Co-ordinating application & network events with business processes to speed business agility.
  - (iii) Aligning network resources to applications to meet business objectives to provide a competitive differentiation.
- ② It is a framework that enables business to build an intelligent enterprise-wide network infrastructure. SONA accomplishes this by separating the network architecture.

#### Benefits of Cisco SONA:-

- (i) Functionality :- Supports the organizational requirements.
- (ii) Scalability :- Supports growth and expansion of organizational tasks.
- (iii) Availability :- Provides necessary services reliable anywhere, anytime.
- (iv) Performance :- Provides desired responsiveness, throughput, and utilization on a per-application basis.
- (v) Manageability :- Provides control, performance monitoring & fault detection.
- (vi) Efficiency :- SONA provides network services and infrastructure with reasonable cost.

There are 3 layers of Cisco SONA : They are:-

(i) The networked infrastructure layer:-

- Where all the IT resources are interconnected across a converged network foundation.
- The objective in these layers is to have connectivity with customer anytime & anywhere.

(ii) The interactive services layer:-

- It enables efficient allocation of resources to applications & business processes delivered through the networked infrastructure.
- The integrated network services layer holds key network infrastructure services, including security, mobility, storage, unified communications, computing, application delivery & identity.

(iii) The application layer:-

- It contains the business applications and collaborative applications that take advantage of efficiencies from the interactive services.
- The objective for customer in this layer is to meet business requirements and achieve efficiencies by leveraging the interactive services layer.