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

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

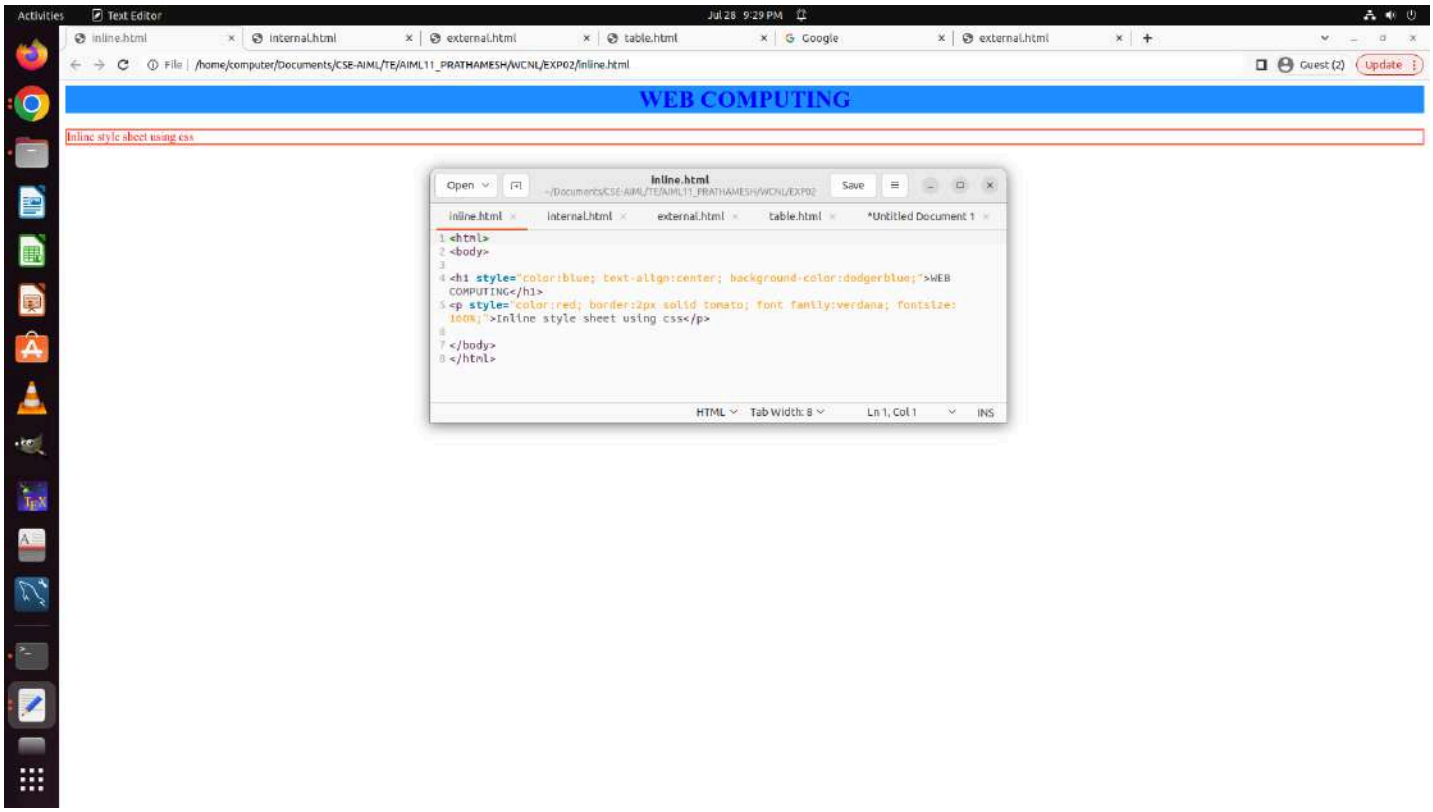
NAME	**CHIKANKAR PRATHAMESH SHIVAJI**
BRANCH	**CSE-(AI&ML)**
ROLL NO.	**AIML11**
SUBJECT	**WEB COMPUTING AND NETWORK LAB**
COURSE CODE	**CSL501**
PRACTICAL NO.	
DOP	
DOS	

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) -

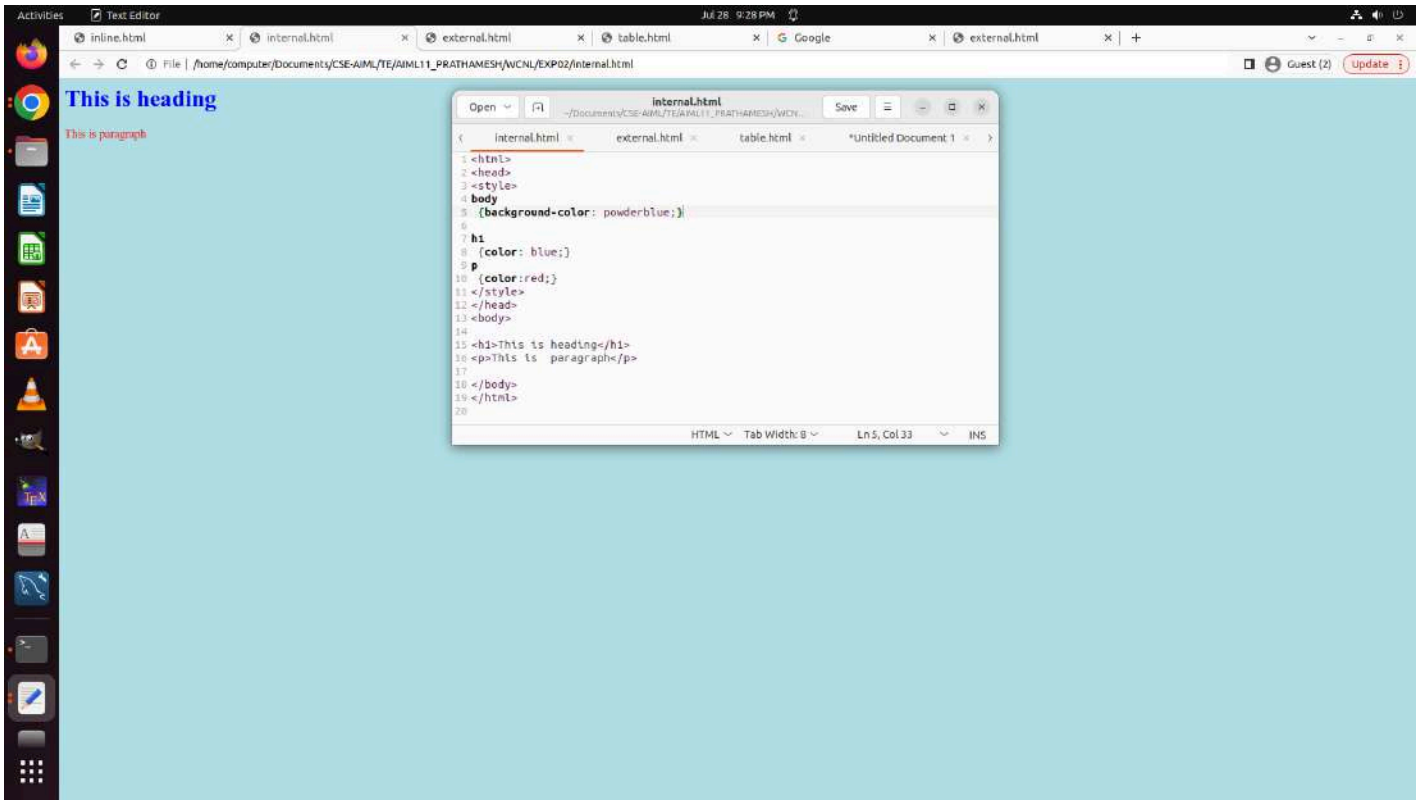


2.internal.html

```
<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

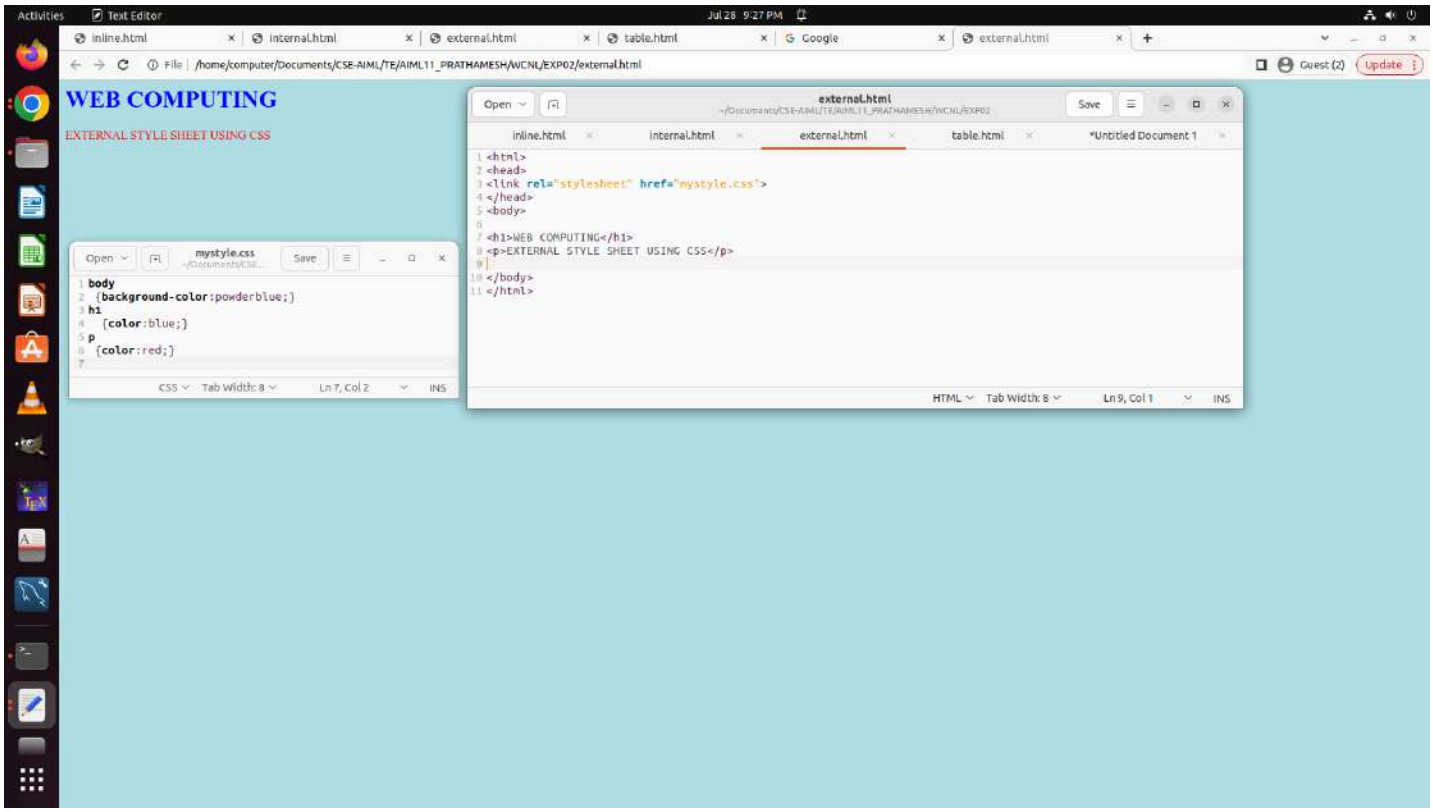
```
<html>
<head>
```

```
<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;}
```

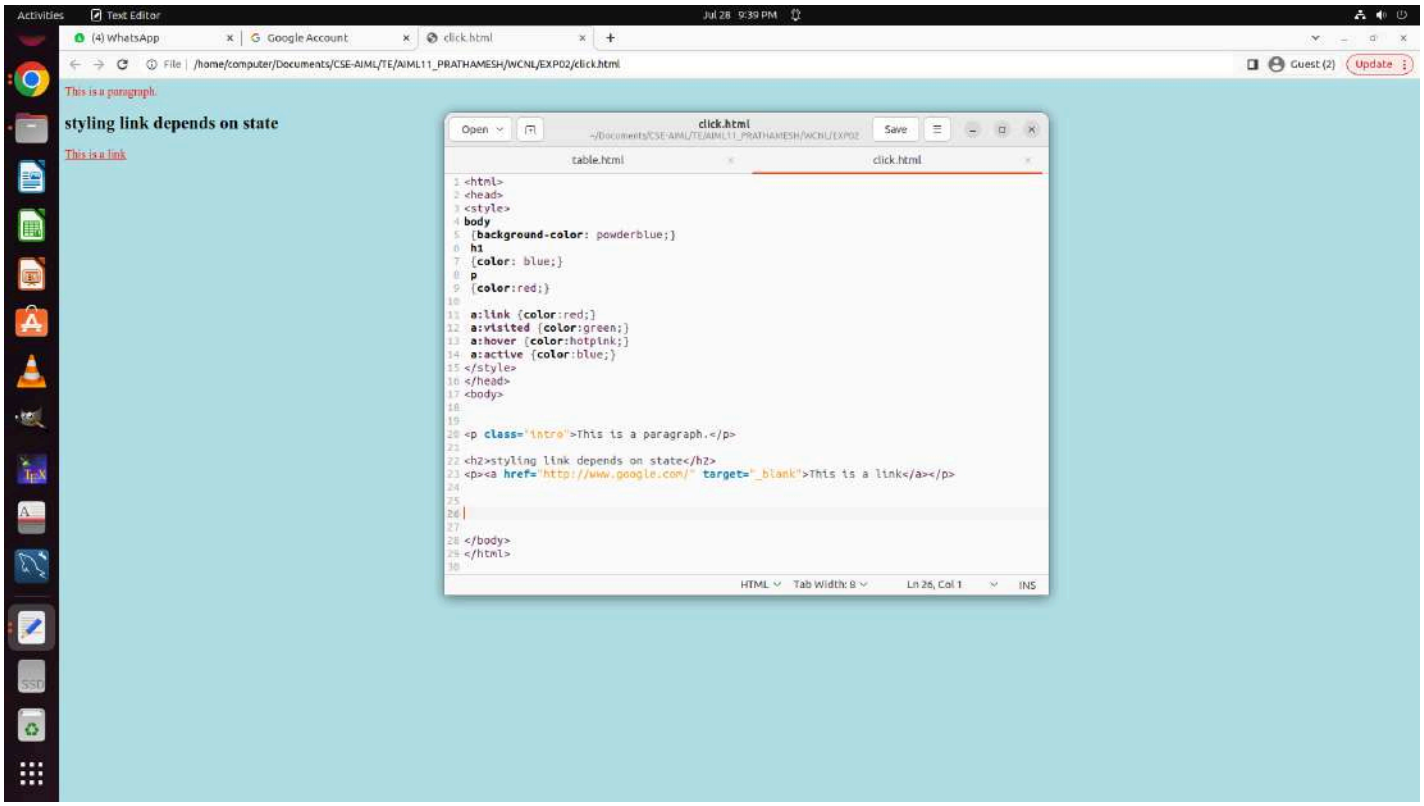
Output (ScreenShot) -

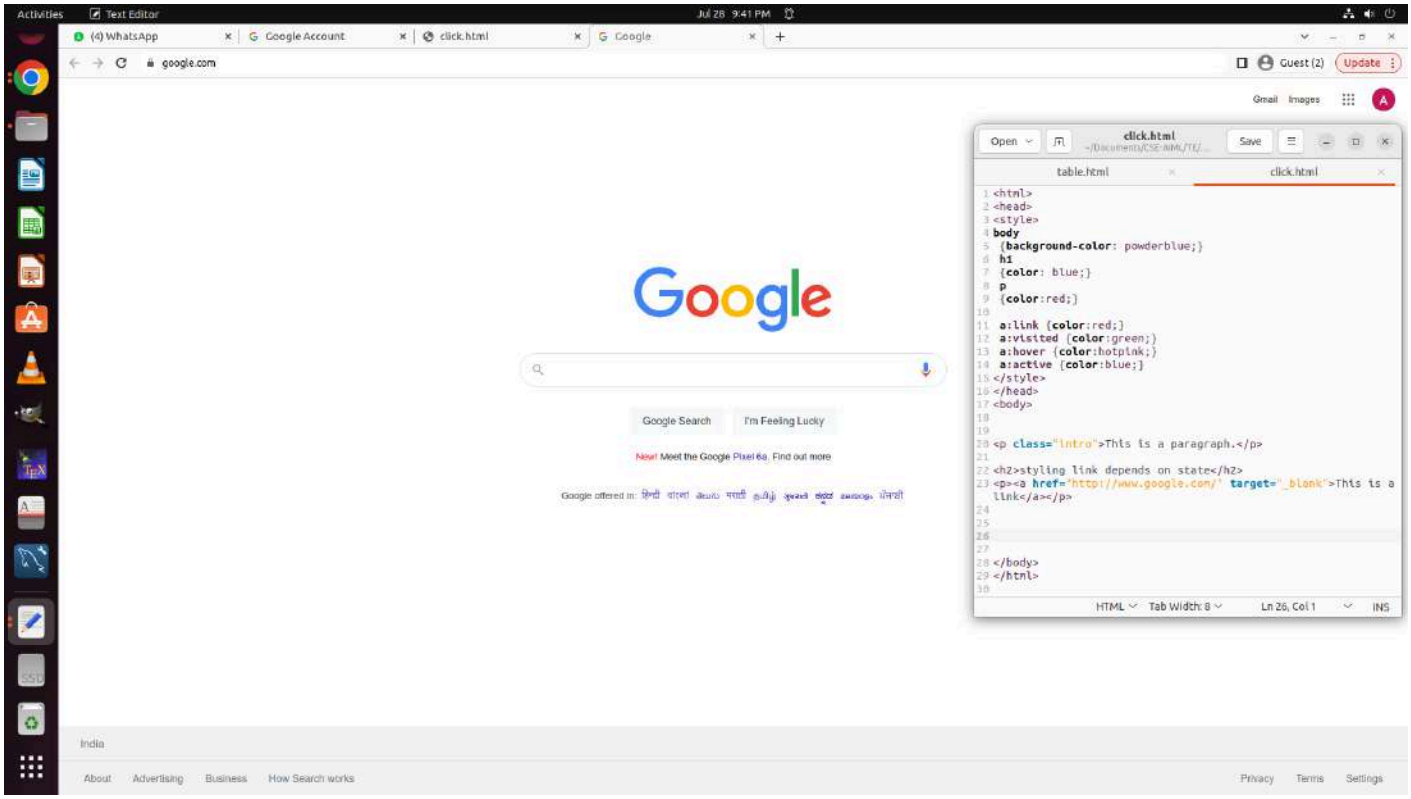


4.click.html

```
<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) -

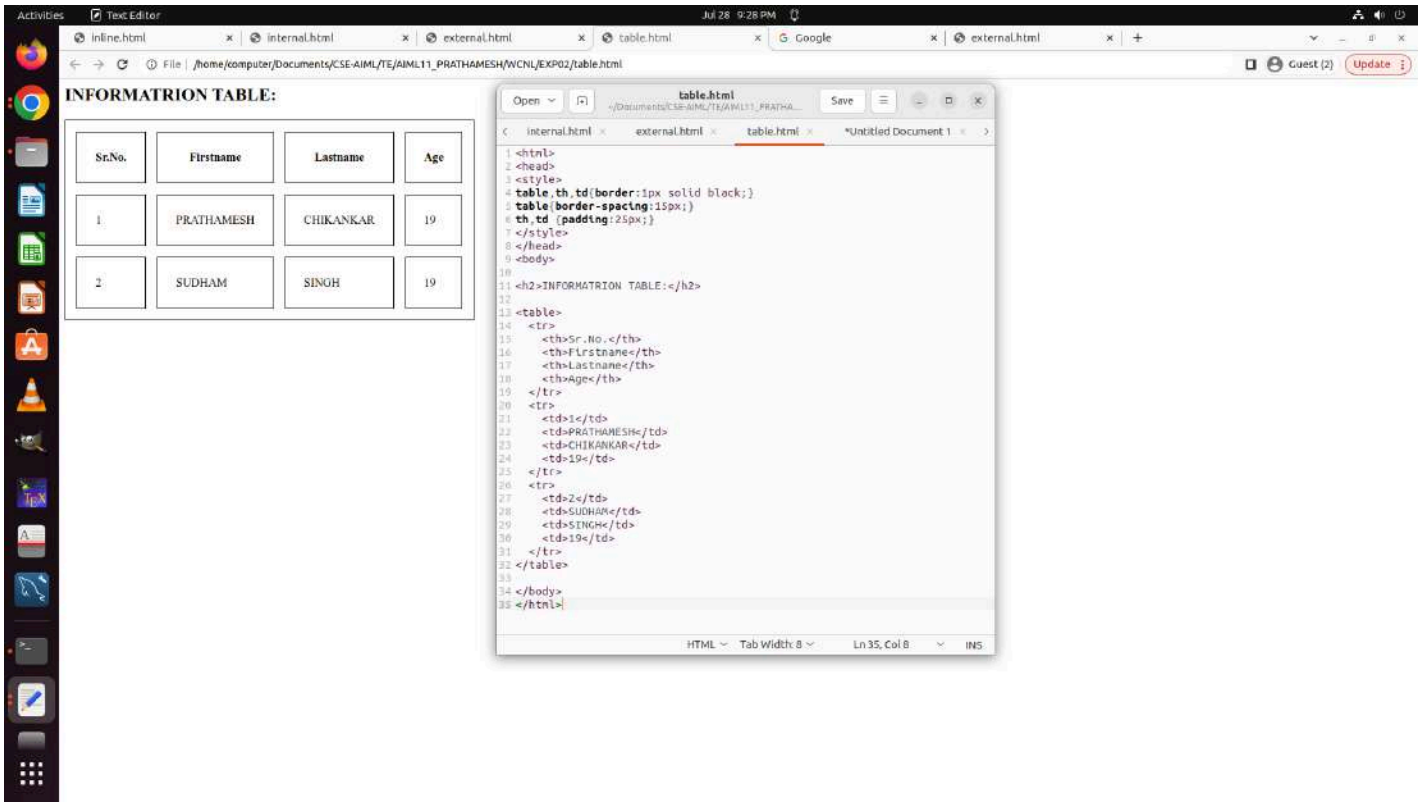




5.table.html

```
<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) -



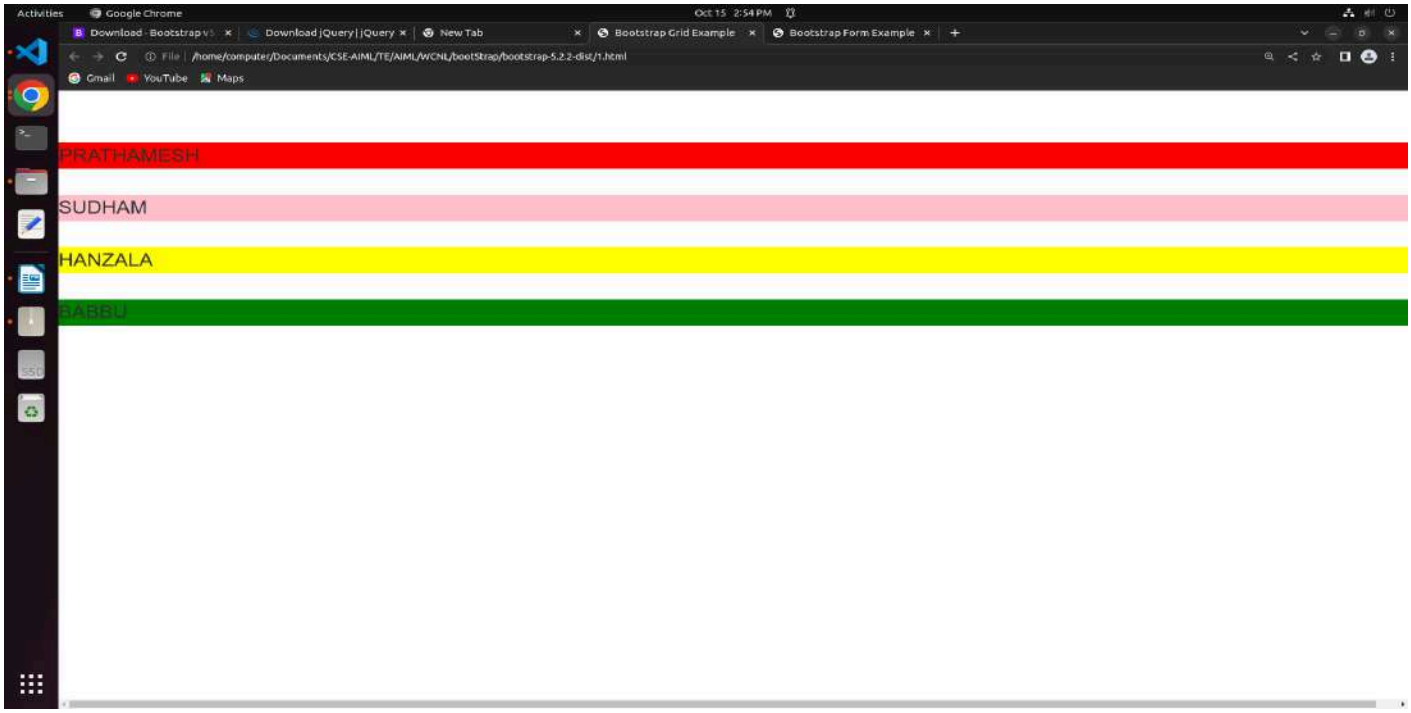
EXPERIMENT NO.03 - BOOTSTRAP

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;">BABBU</div>
      </div>
  </body>
</html>
```

OUTPUT:



2.Bootstrap vertical form

CODE:

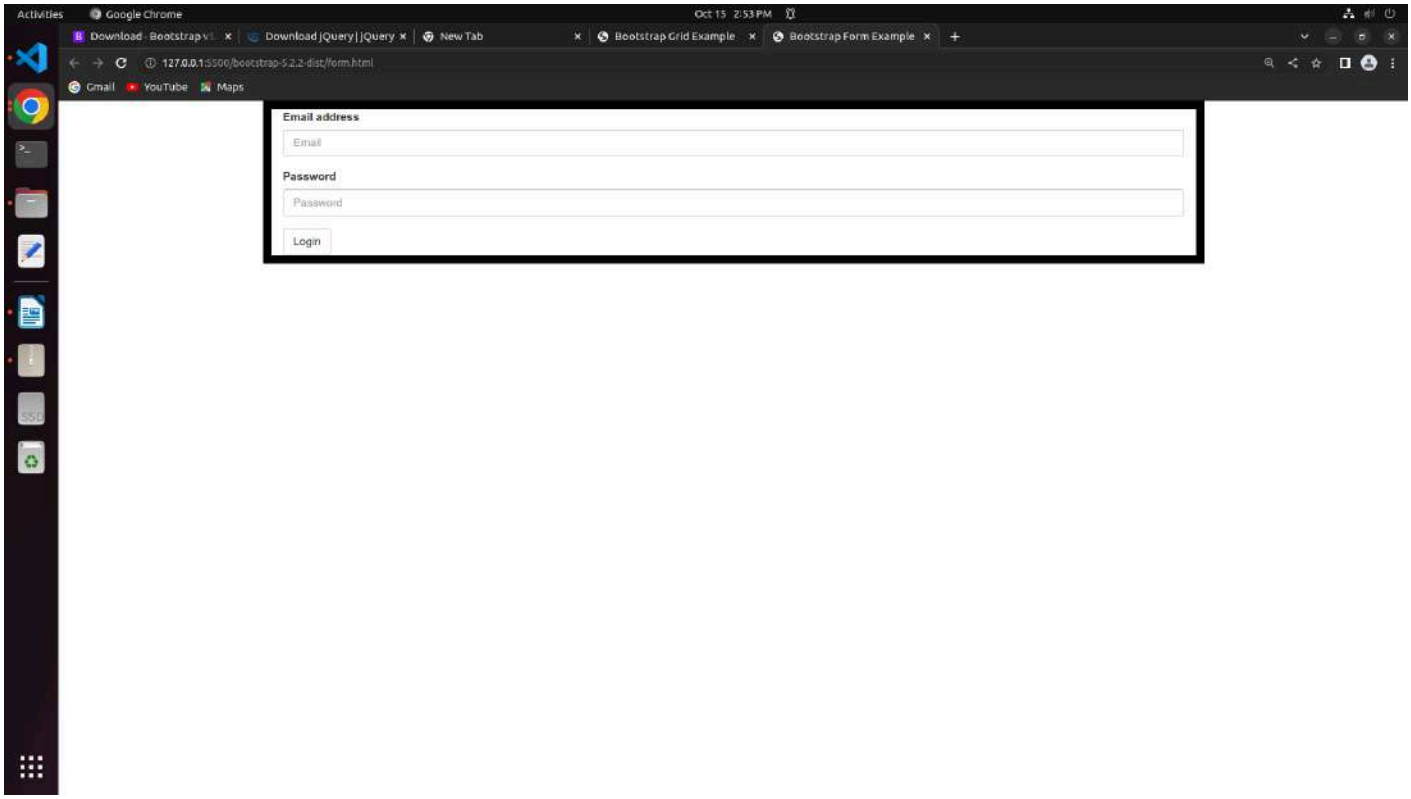
formStyle.css

```
.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:

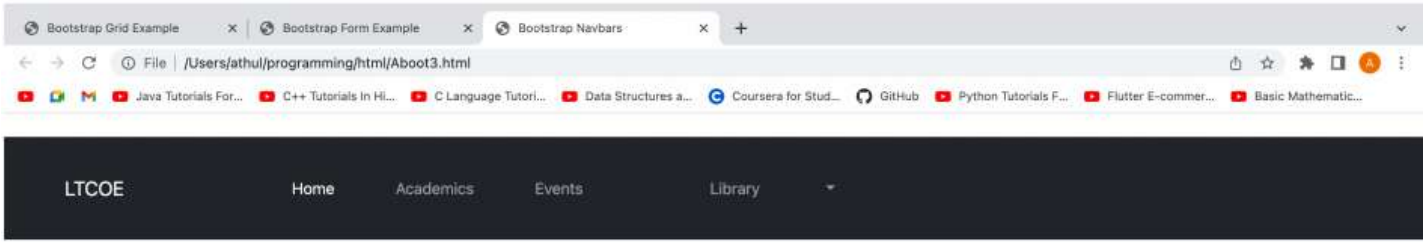
4.Bootstrap Navbars

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/
azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
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>
</html>
 <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 -->
 <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+Jc
Xn/
tWtIaxVXM" crossorigin="anonymous"></script>
 </body>
</html>
```

OUTPUT:



5.Bootstrap Breadcrumbs

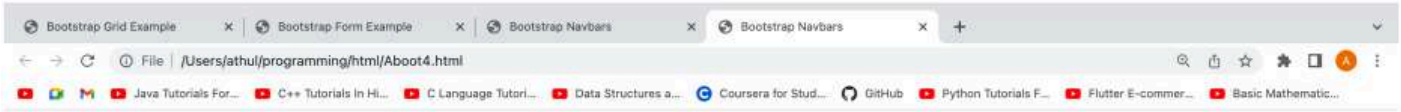
CODE:

```
<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/
azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
```

```
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 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 -->
 <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+Jc
Xn/
tWtIaxVXM" crossorigin="anonymous"></script>
```

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

OUTPUT:



Home

Home / Library

Home / Library / Data

EXPERIMENT NO.04 - JAVASCRIPT

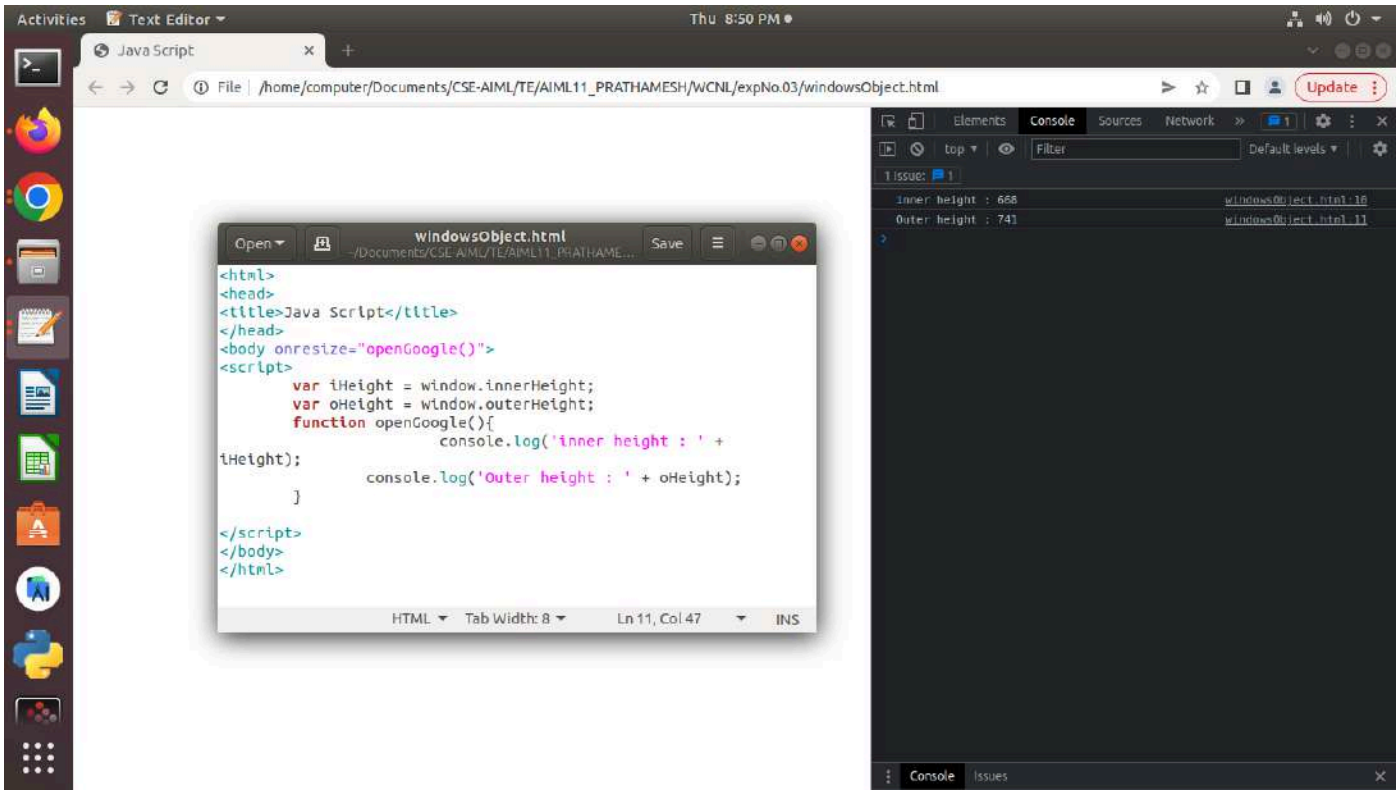
4a-WindowOjbect

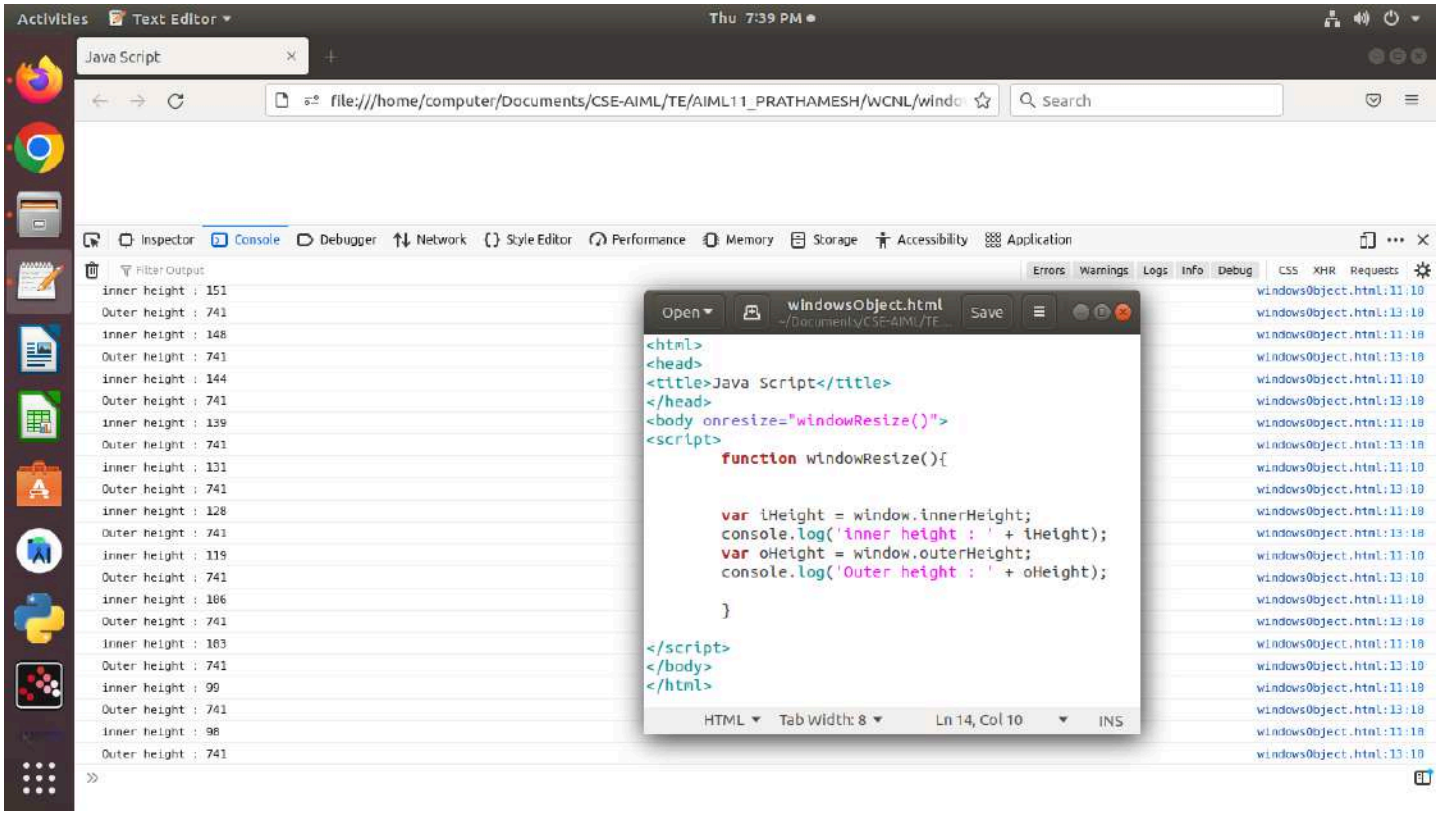
1.windowObject

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:





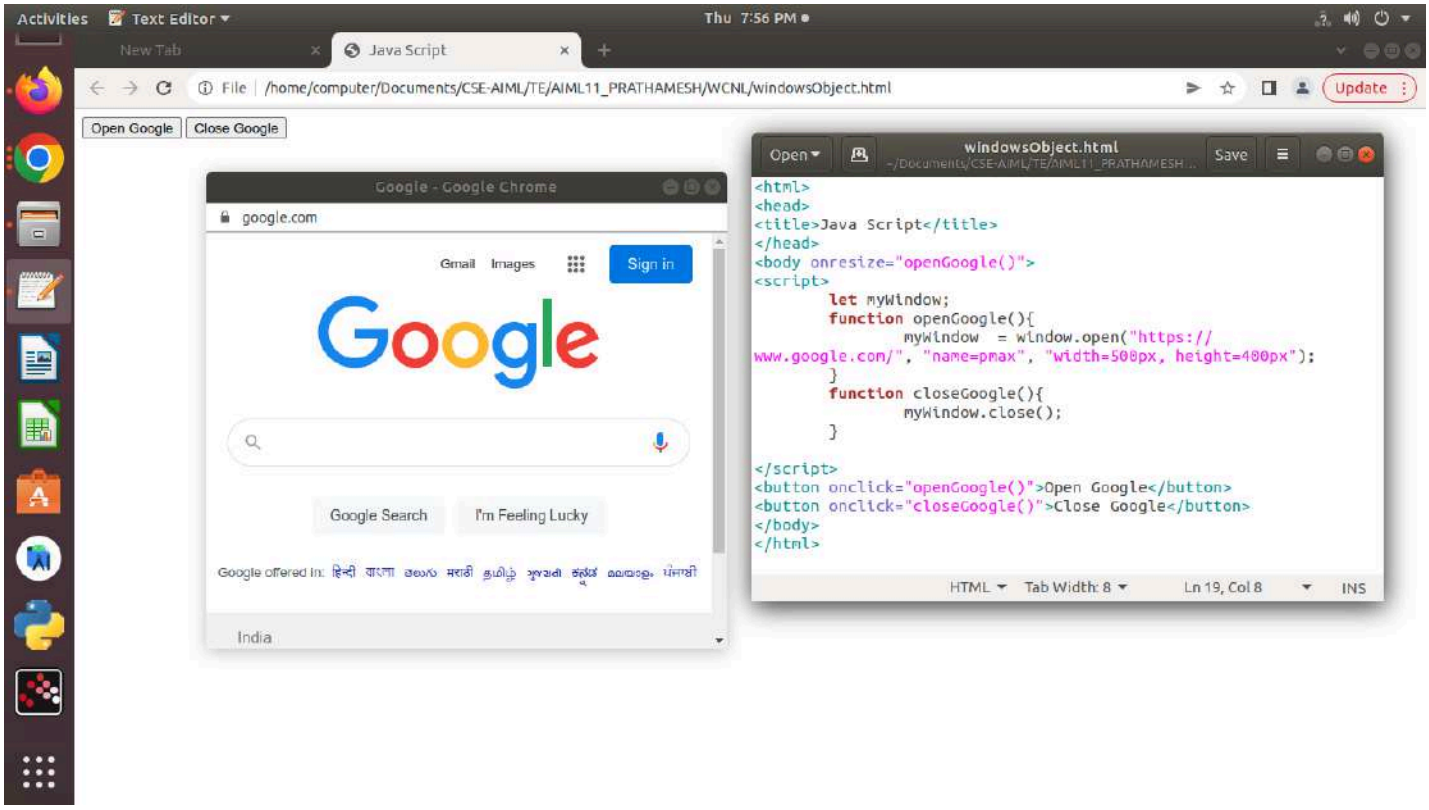
2.openCloseGoogle

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:



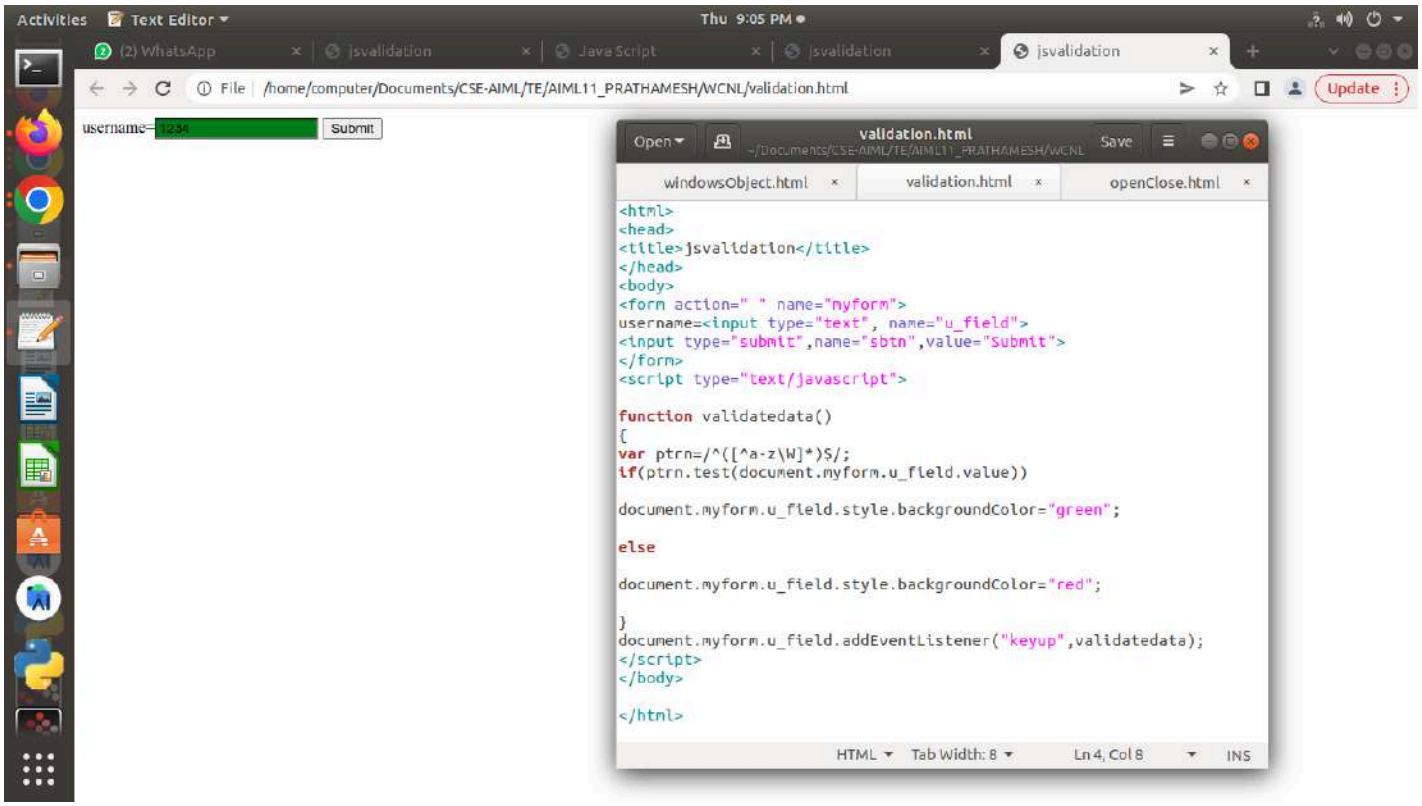
3.validation

CODE:validation.html

```
<html>
<head>
<title>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-z\W]*)$/;
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:



4b-Loops

1.CODE:FOR IN LOOP

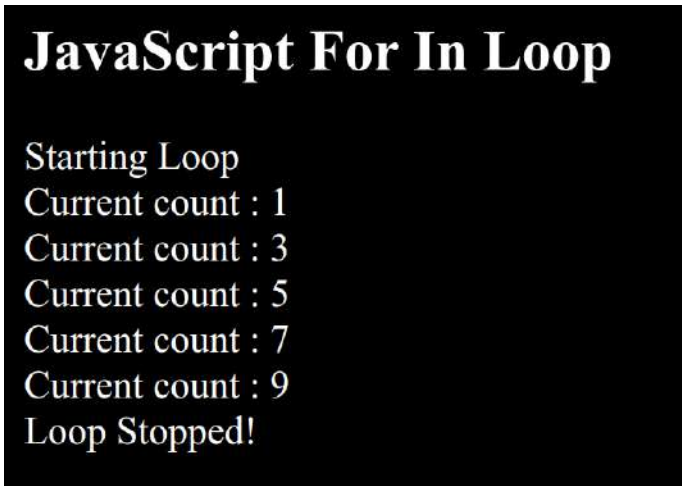
```
<html>
<head>
	<title>For In Loops</title>
	<style>
		html {
			background-color: black;
			color: white;
		}
	</style>
</head>
<body>
	<h2>JavaScript For In Loop</h2>
	<script>
		let student = {
			lname: "Prathamesh",
			fname: "PC",
			age: 20
		};
		let txt = "";
		student["email"] = "prathus@gmail.com";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

OUTPUT:

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:

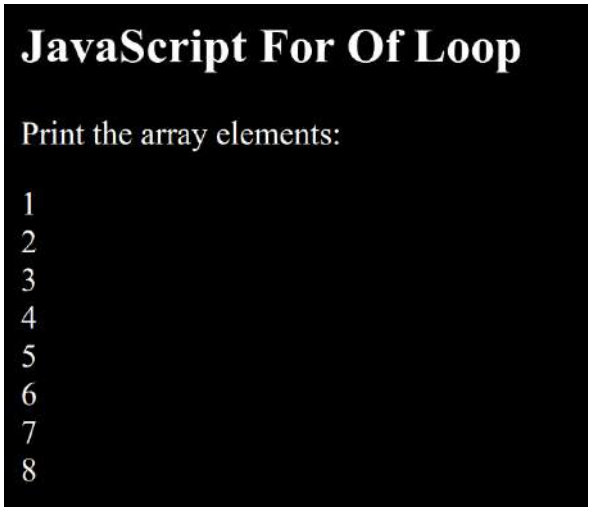


3.CODE:FOR UP LOOP

```
<!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:

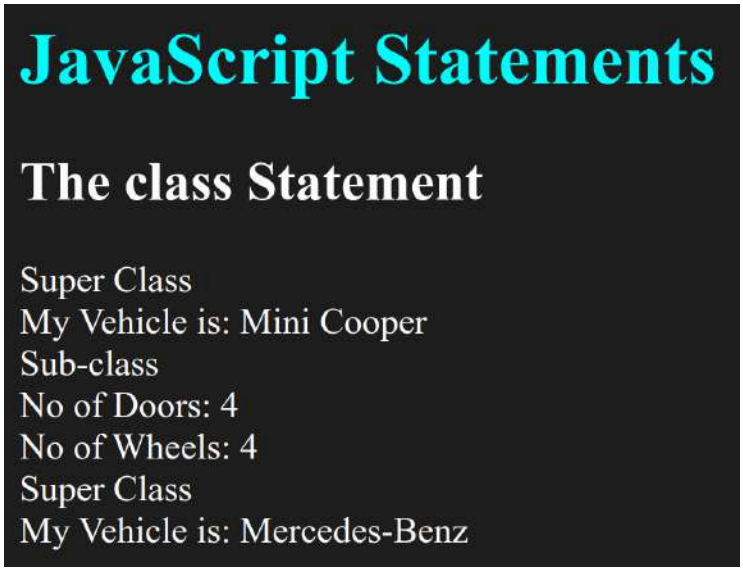


4.CODE:CLASS:

```
<!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>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:



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>
```

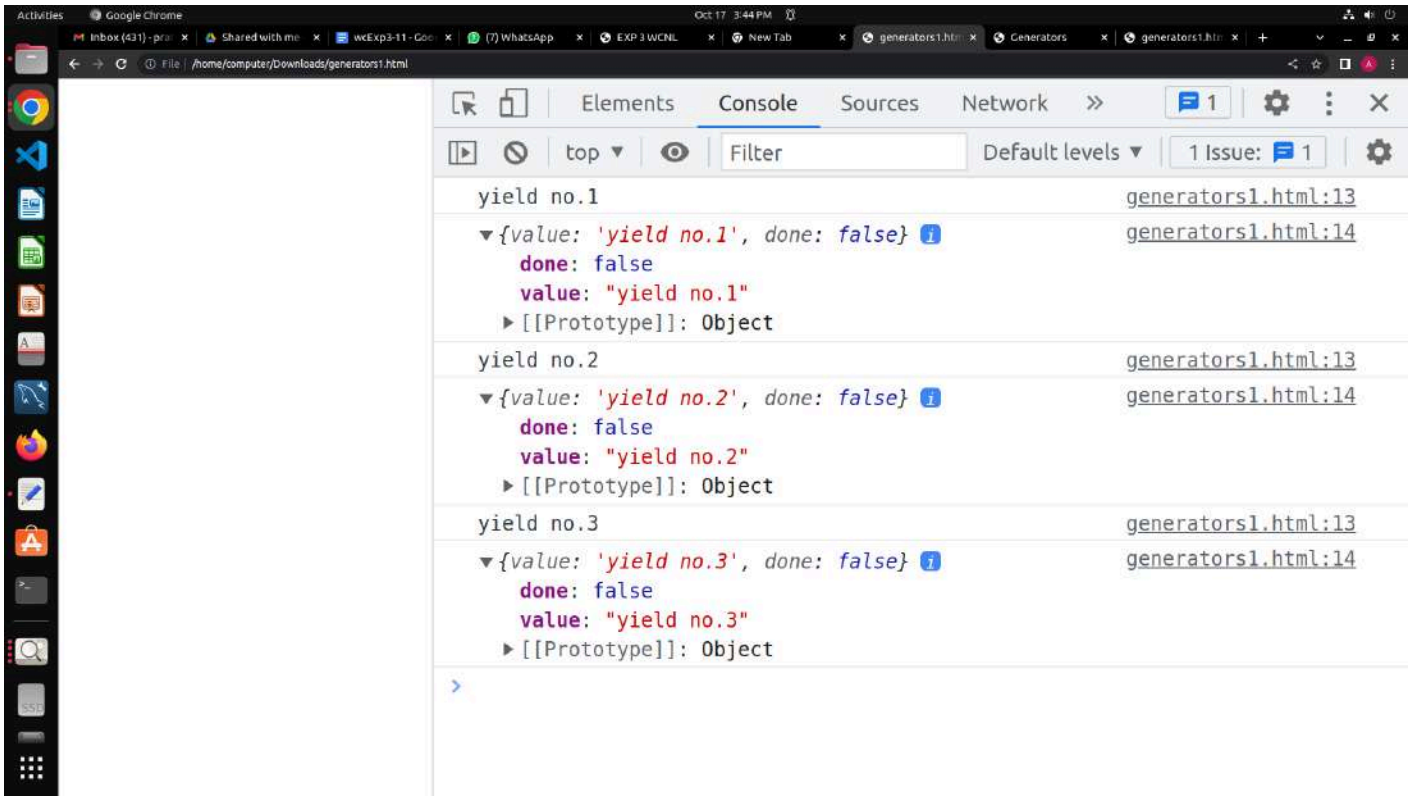
OUTPUT:

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>
```

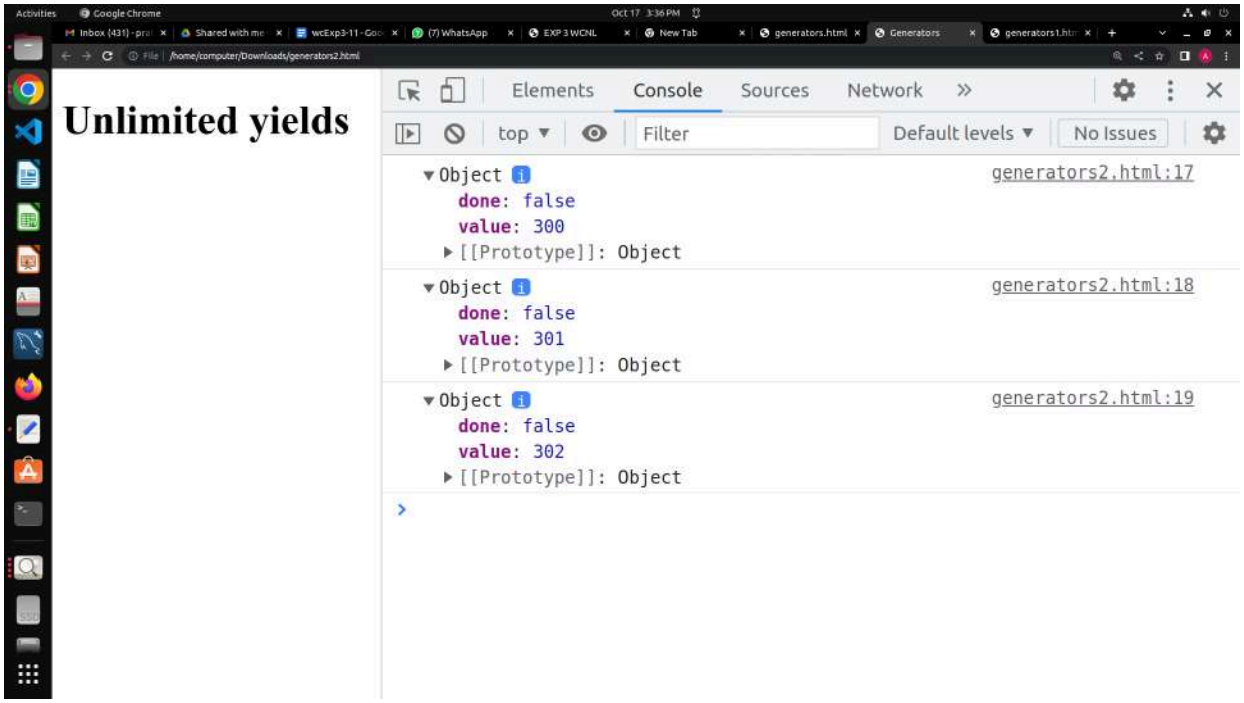
OUTPUT:



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:



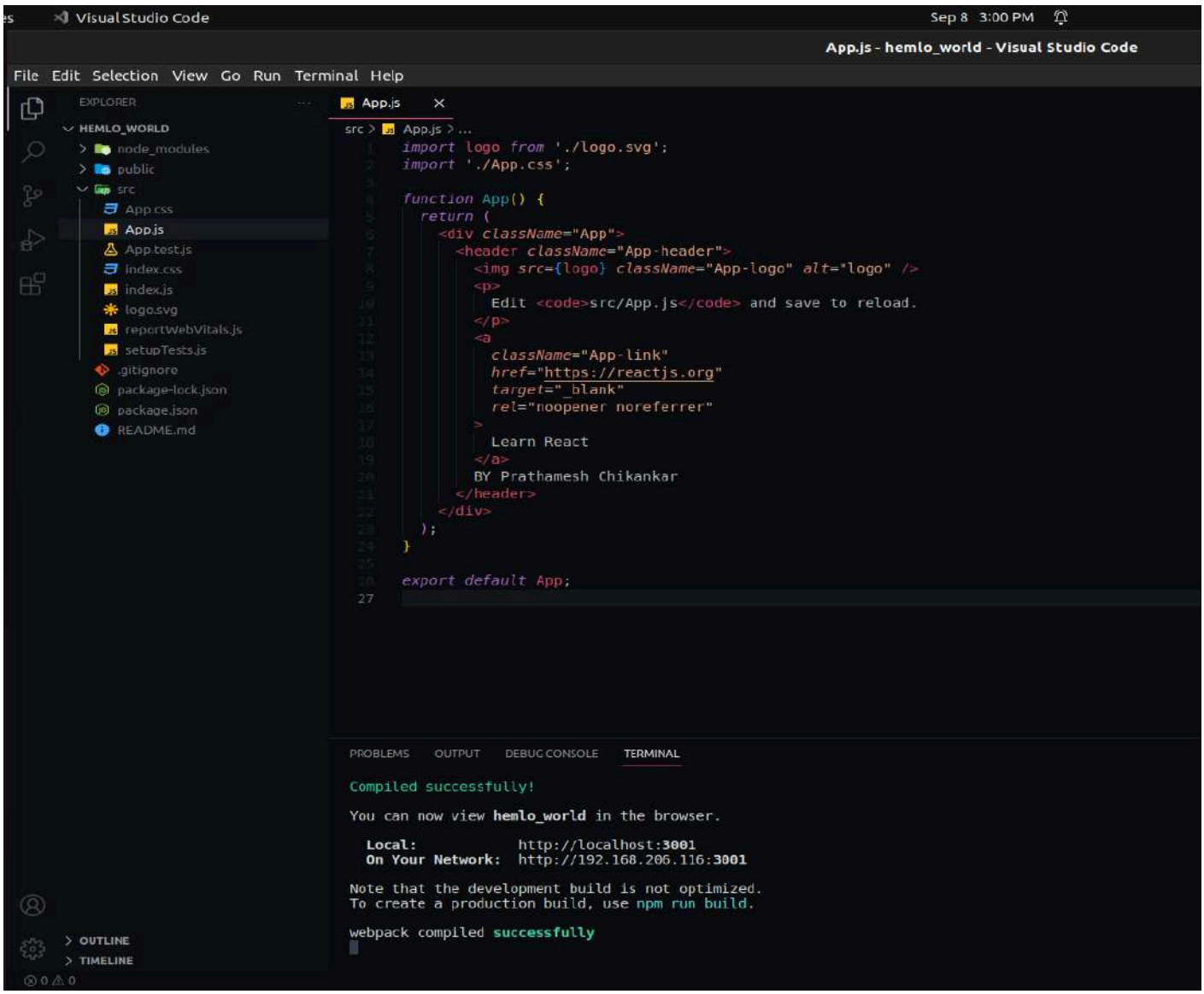
EXPERIMENT NO.05 - REACT.JS

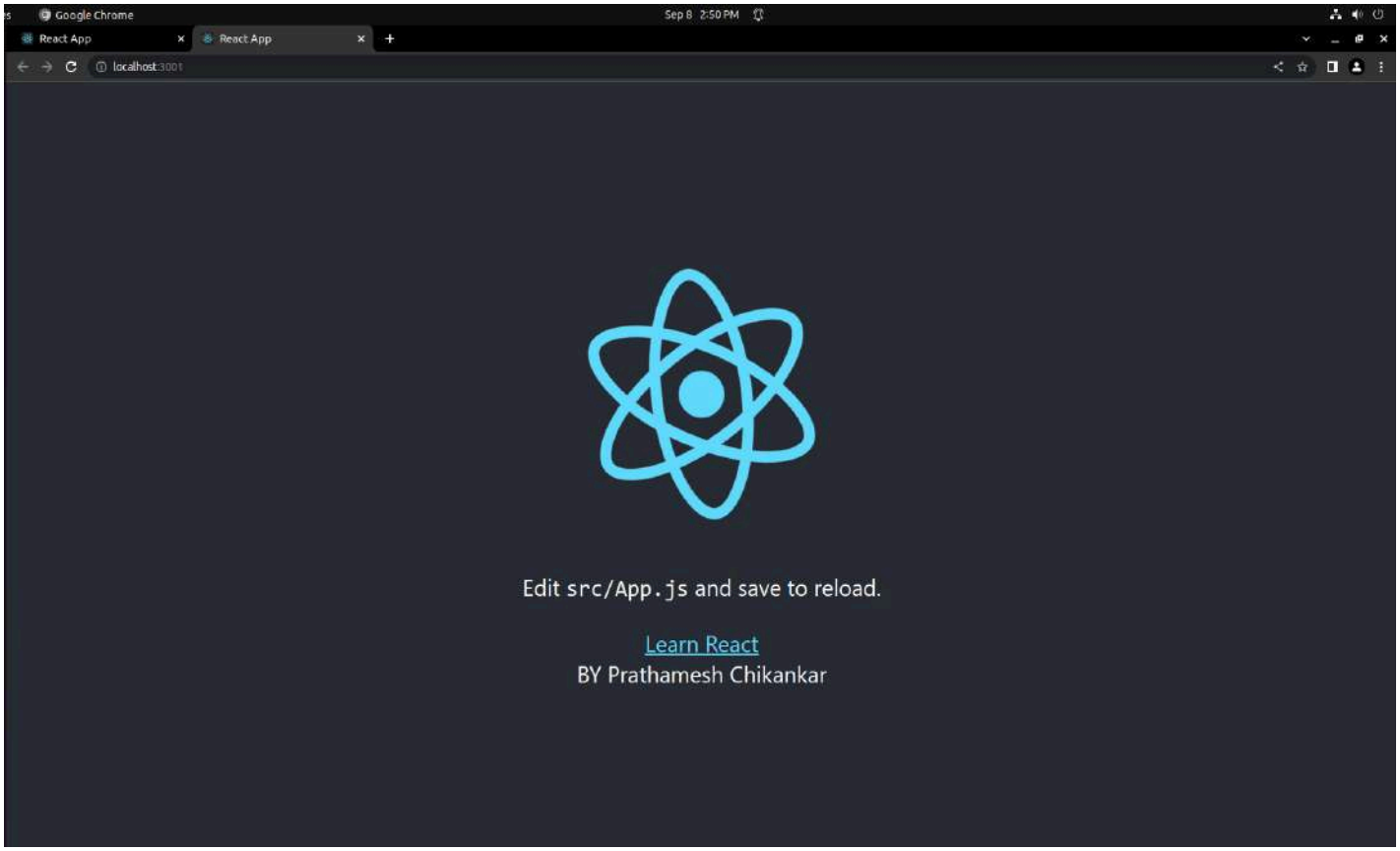
1.Installation

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:





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;
```

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 assync.js > ...

```
var fs = require("fs");

fs.readFile('file1.txt', function (err, filedata) {
    if (err) return console.error(err);
    console.log(filedata.toString());
});
console.log("End of Program execution");
```

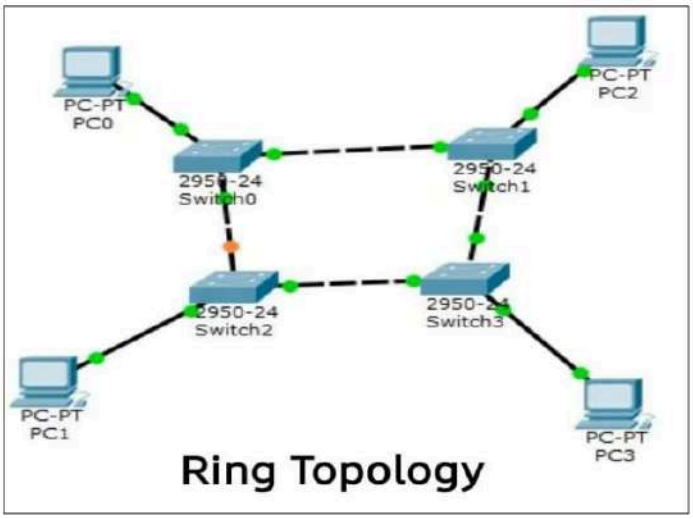
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

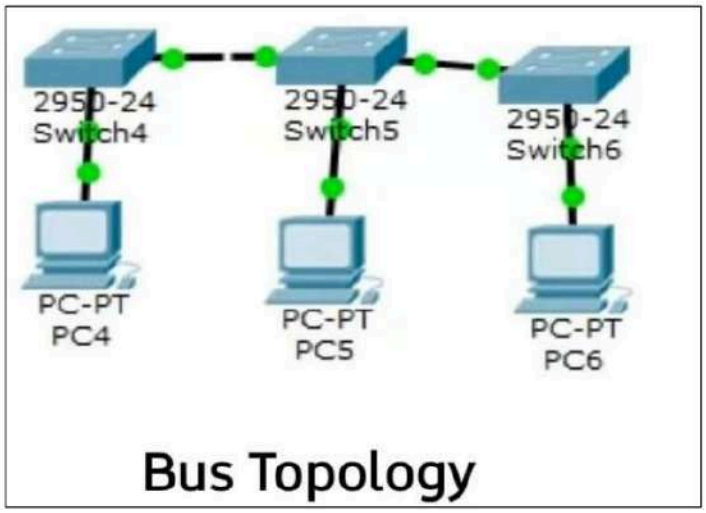
```
PS C:\VS Code\WC\node\src> node sync.js
Hello!!!
Code for reading a file synchronously (blocking code) in Node.js.
End of Program execution
```

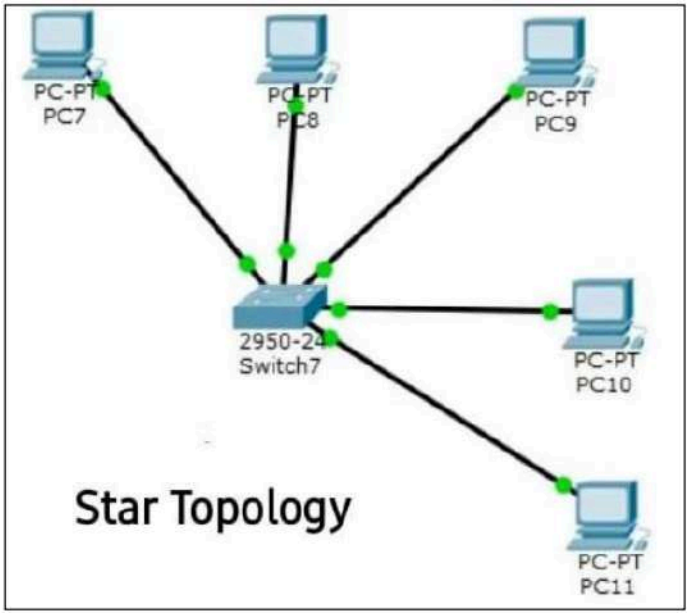
EXPERIMENT NO.07

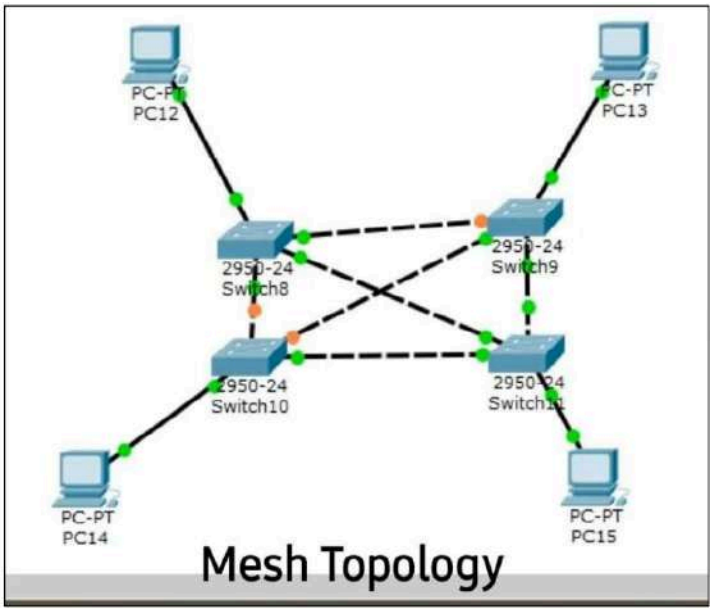
OUTPUT:

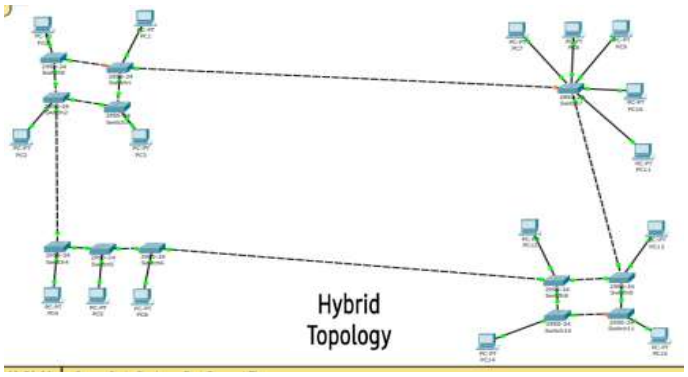
TOPOLOGIES:



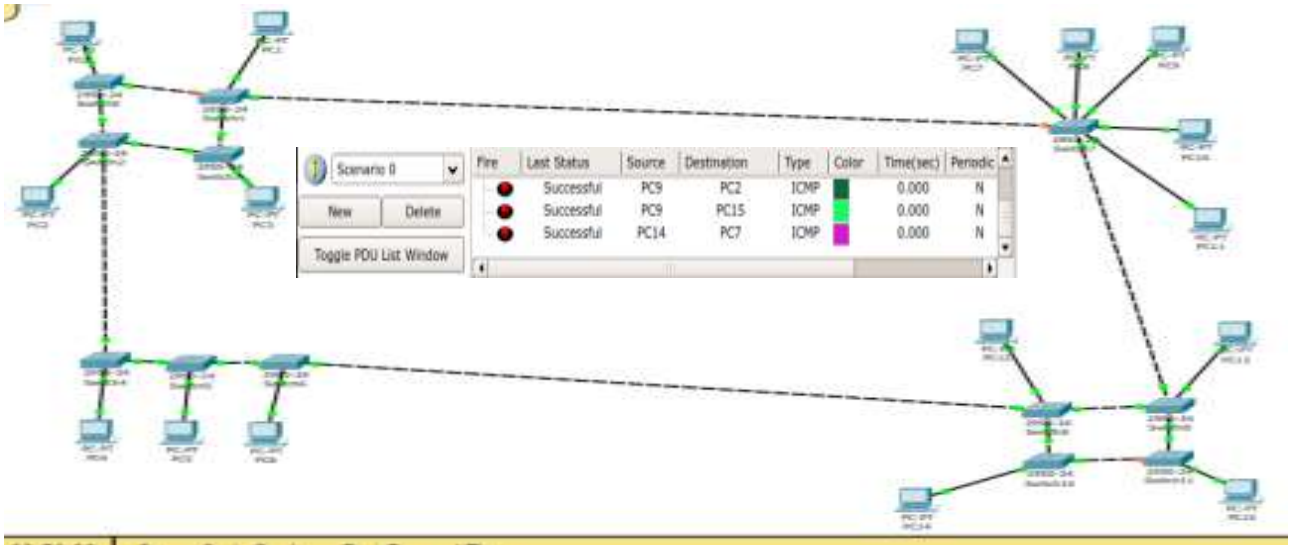


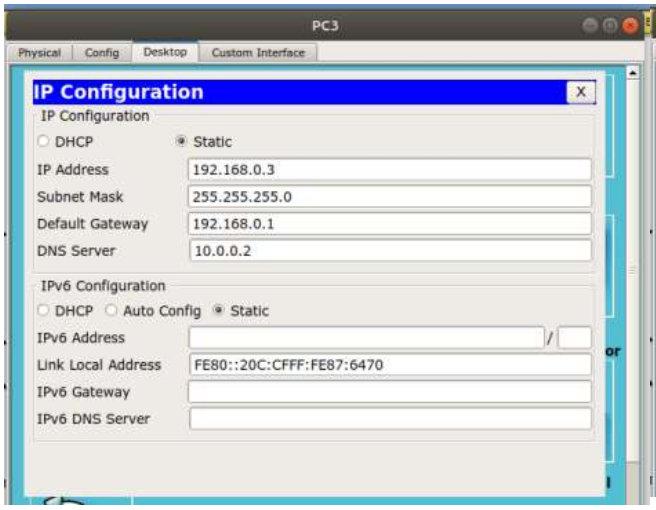


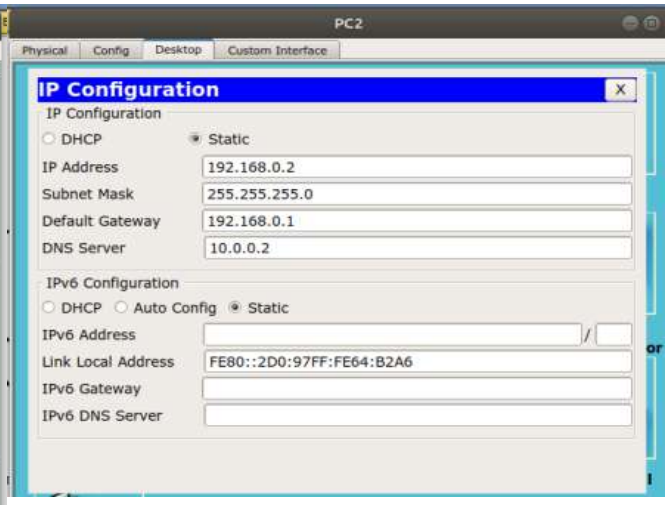


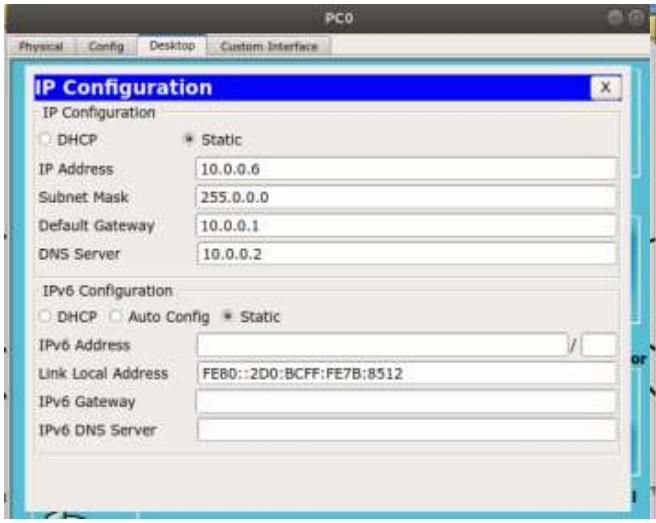


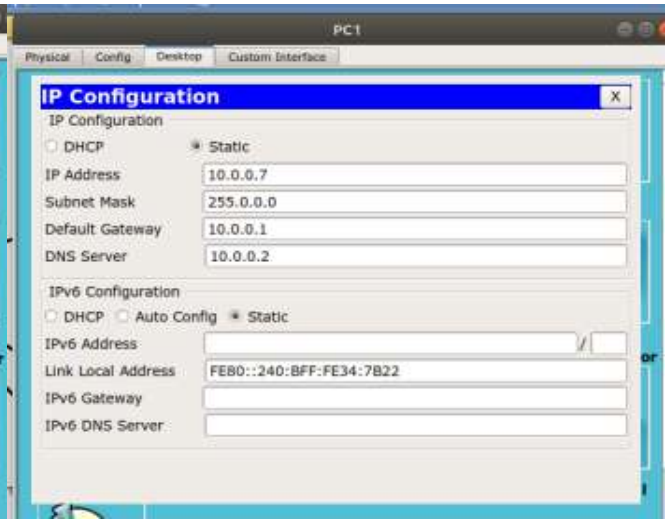
IP setup and its configuration to a static routing :



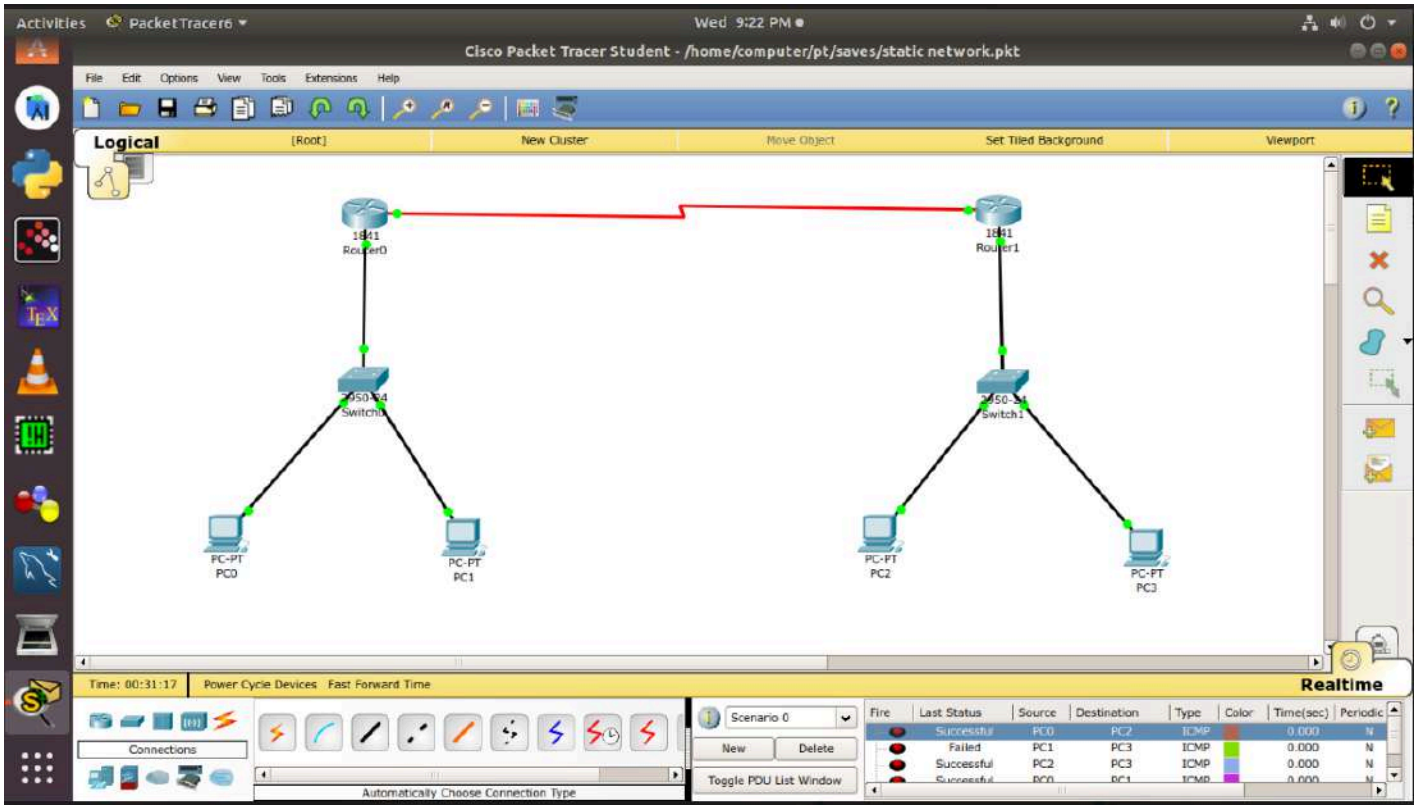


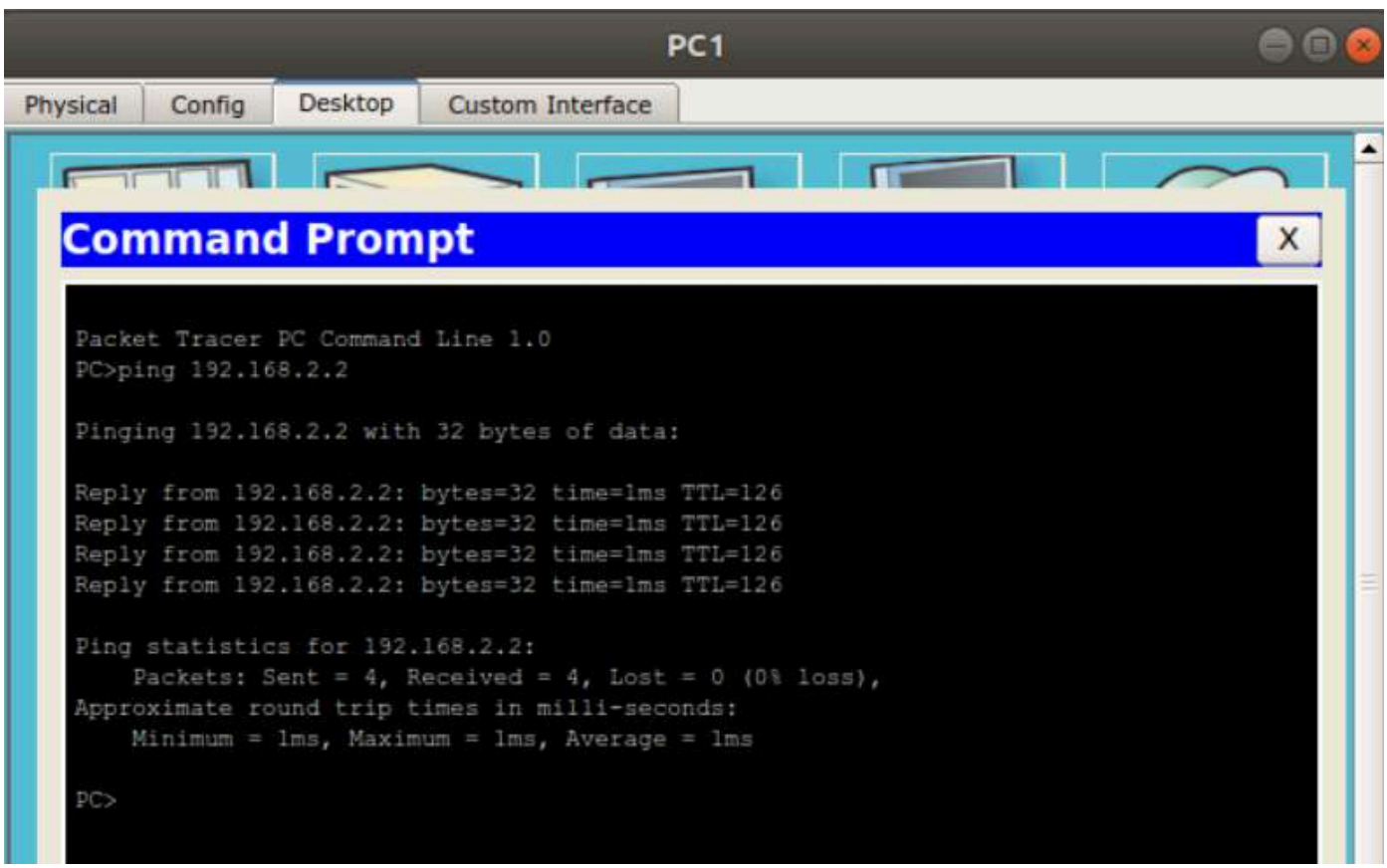






STATIC ROUTING:

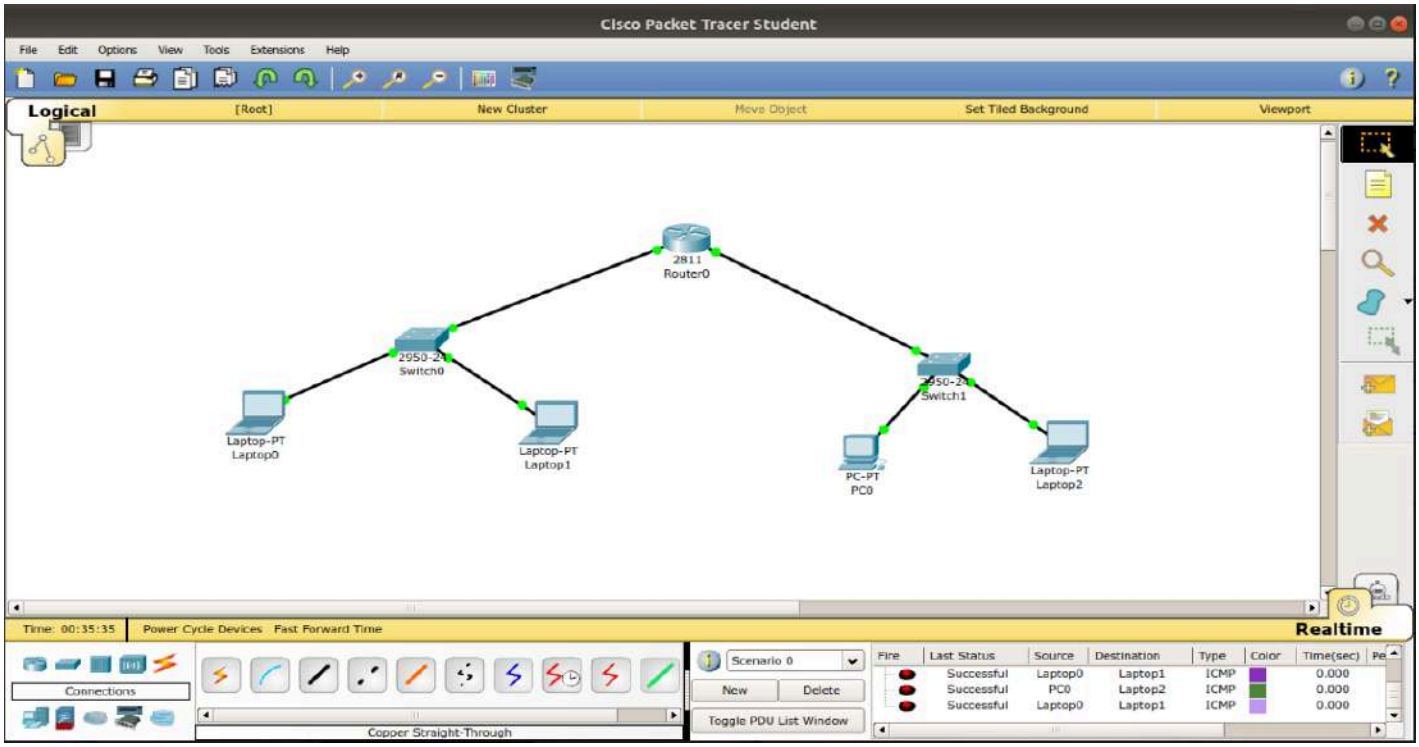


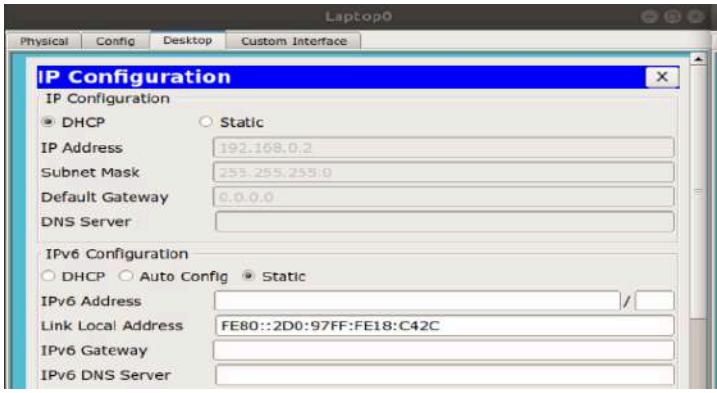


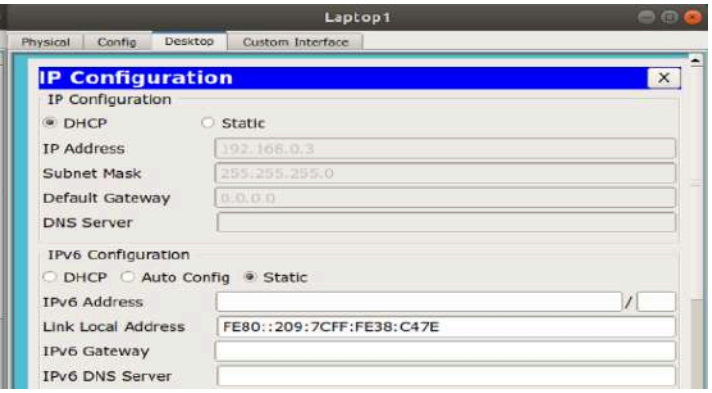
EXPERIMENT NO.08

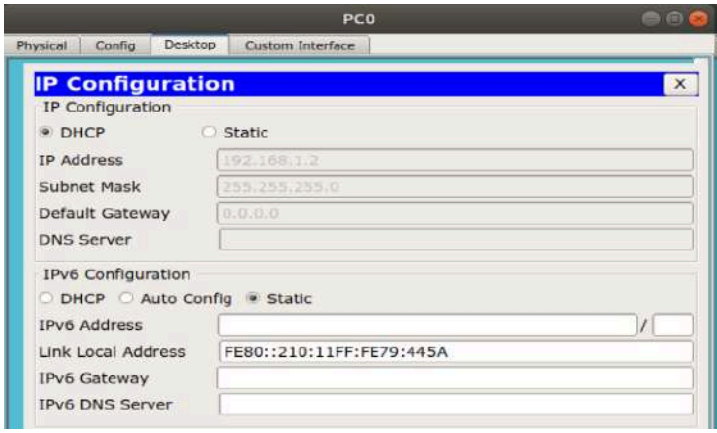
OUTPUT:

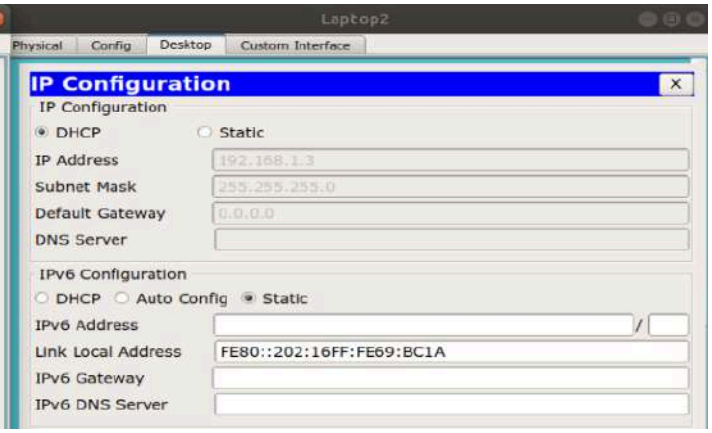
CONFIGURATION OF DHCP SERVER







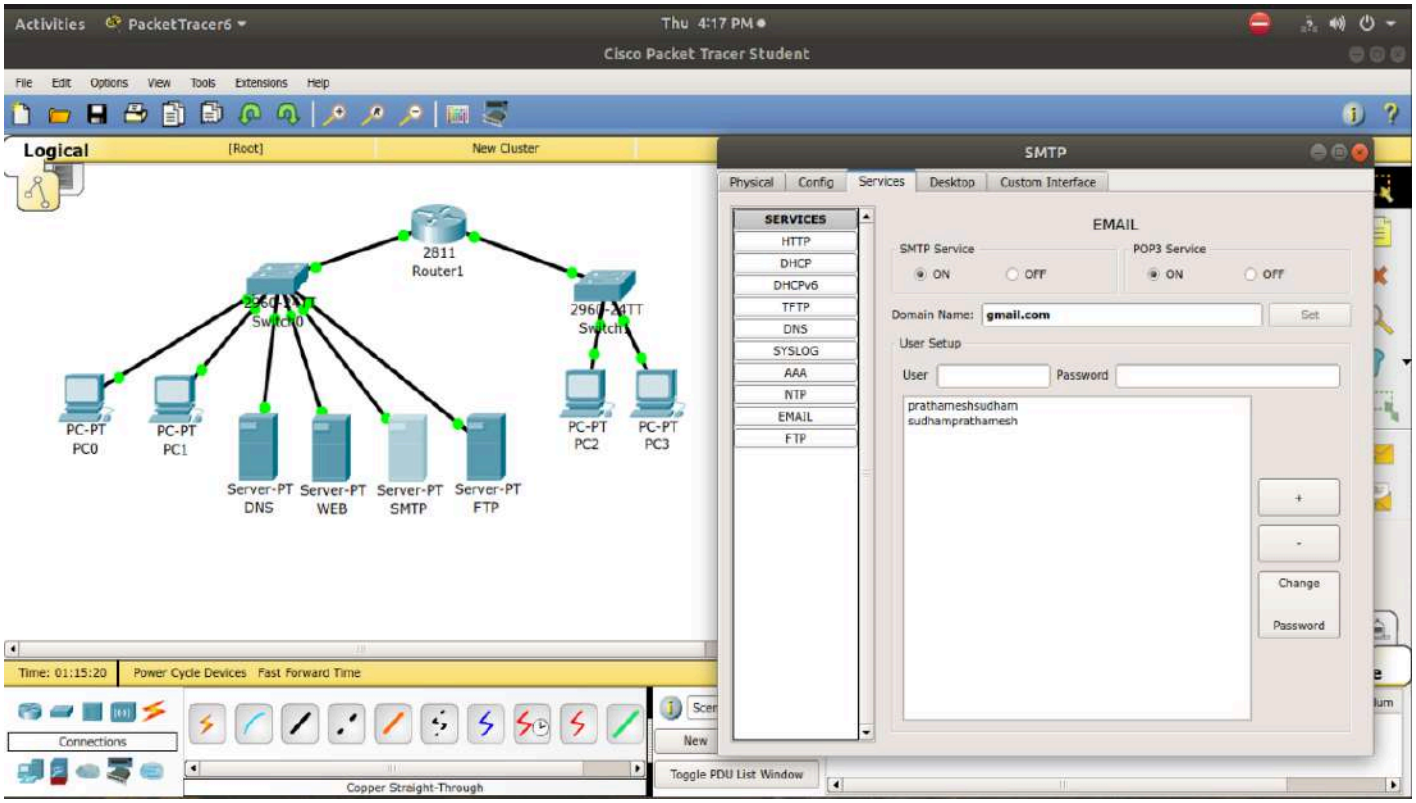


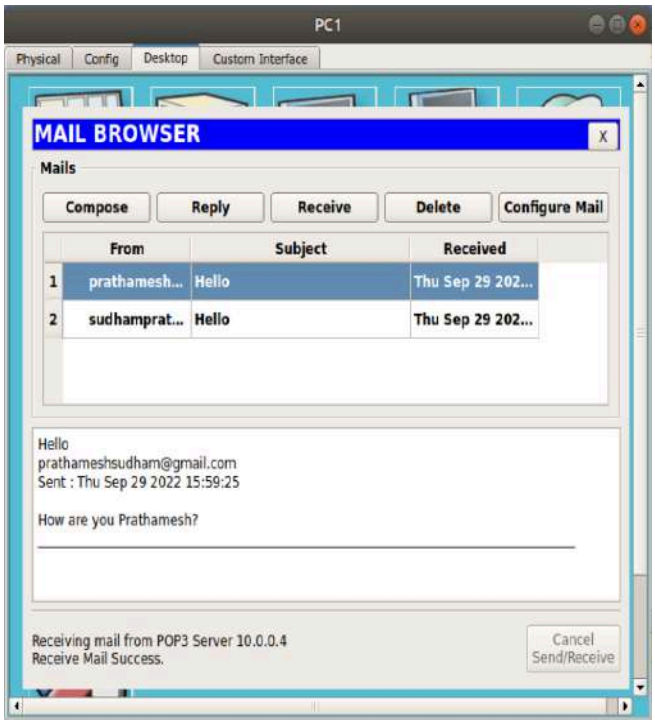


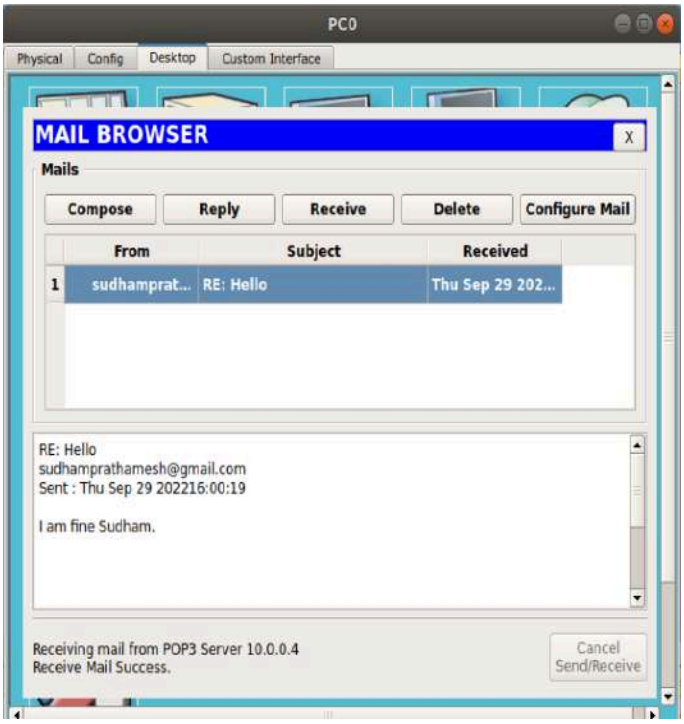
EXPERIMENT NO.09

OUTPUT:

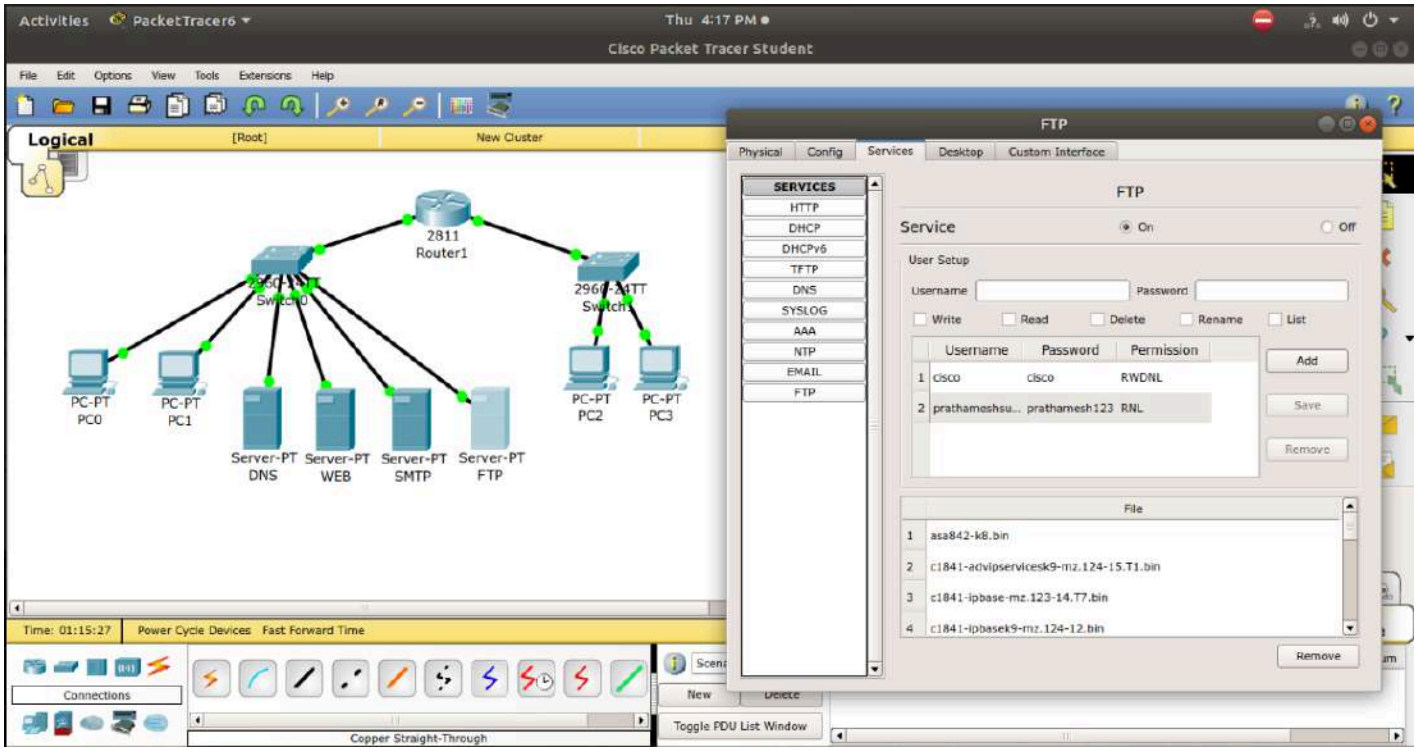
SMTP SERVER-

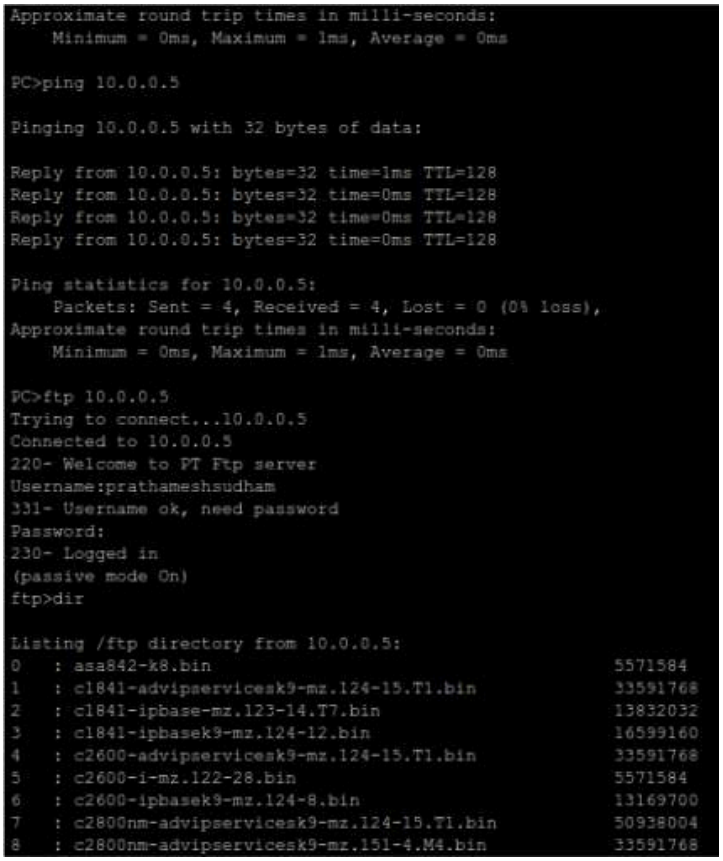




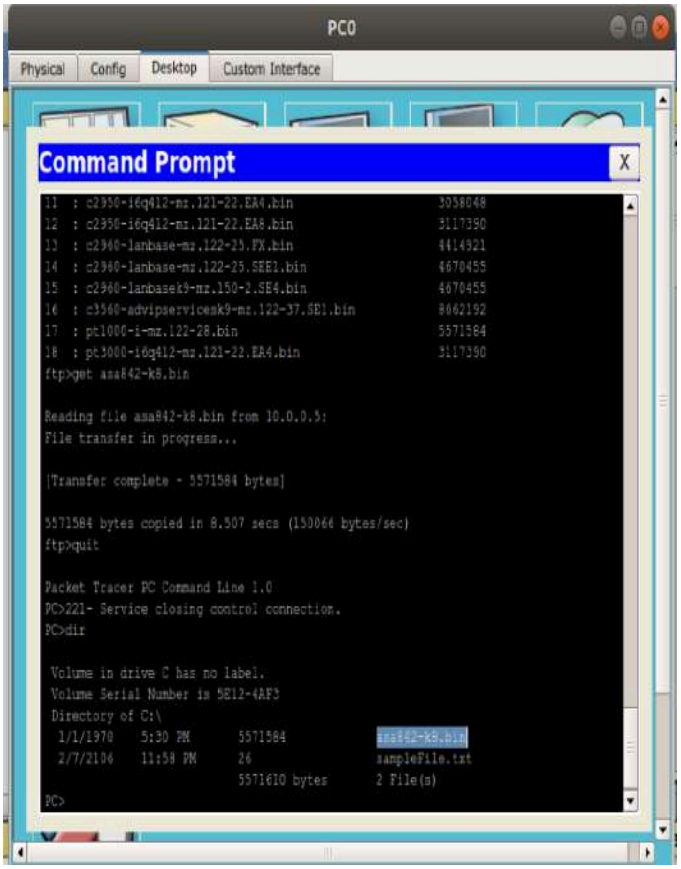


FTP SERVER-





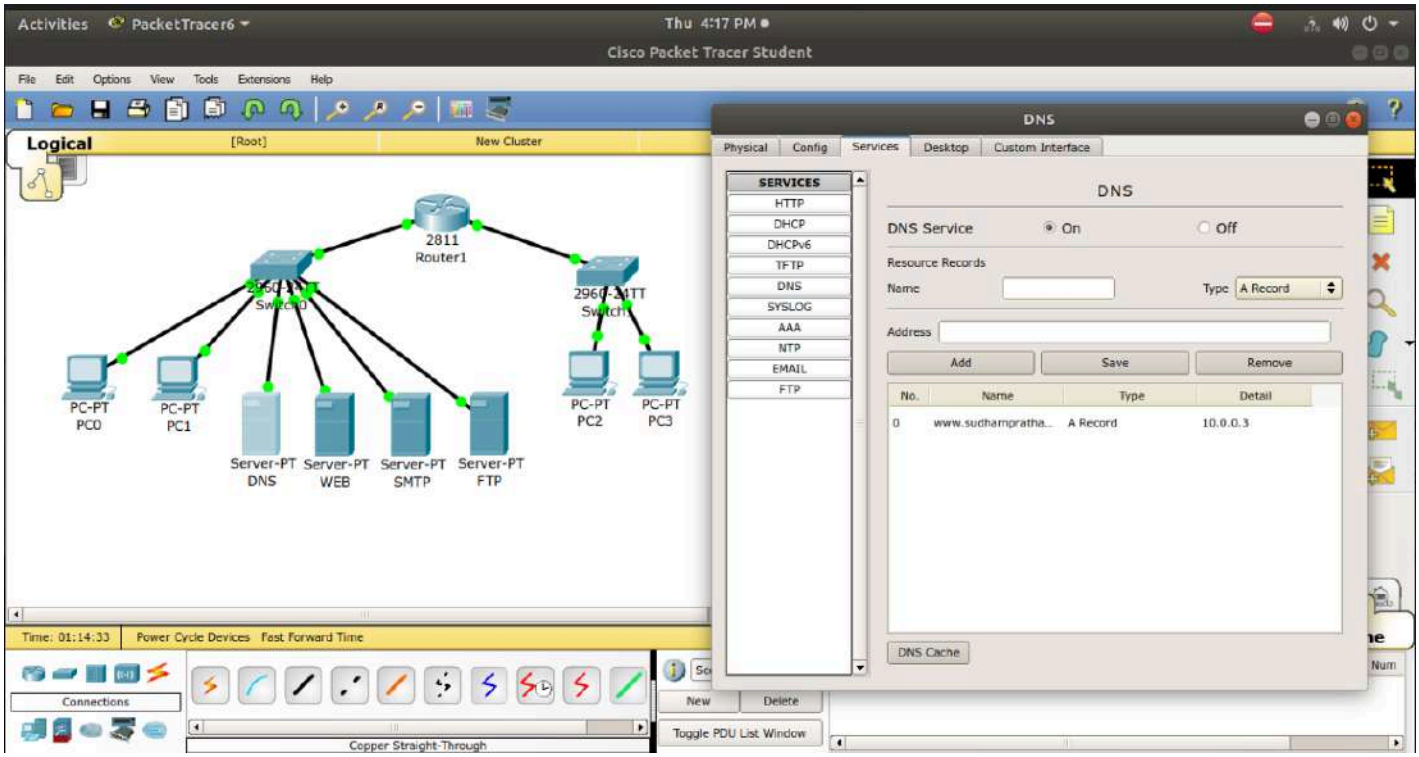
```
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

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-advipservicesk9-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-advipservicesk9-mz.124-15.T1.bin         33591768
5   : c2600-i-mz.122-28.bin                          5571584
6   : c2600-ipbasek9-mz.124-8.bin                    13169700
7   : c2800nm-advipservicesk9-mz.124-15.T1.bin       50938004
8   : c2800nm-advipservicesk9-mz.151-4.M4.bin        33591768
```

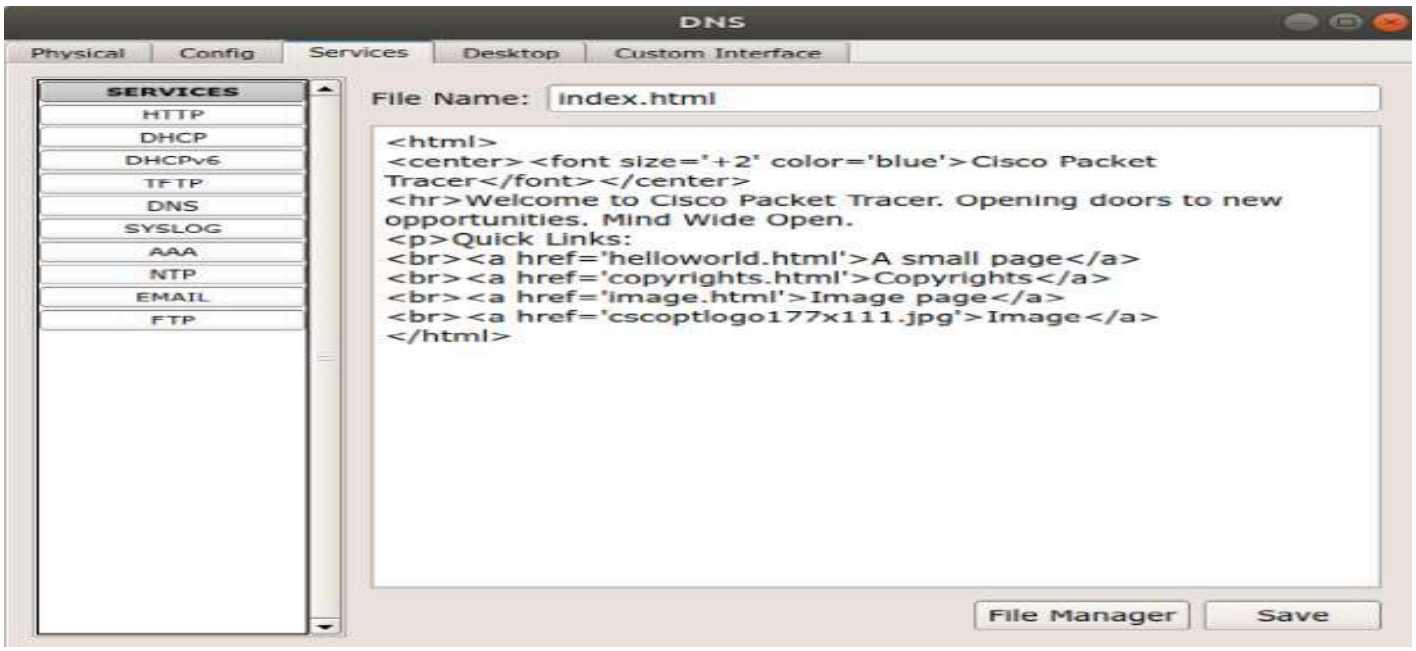


EXPERIMENT NO.10

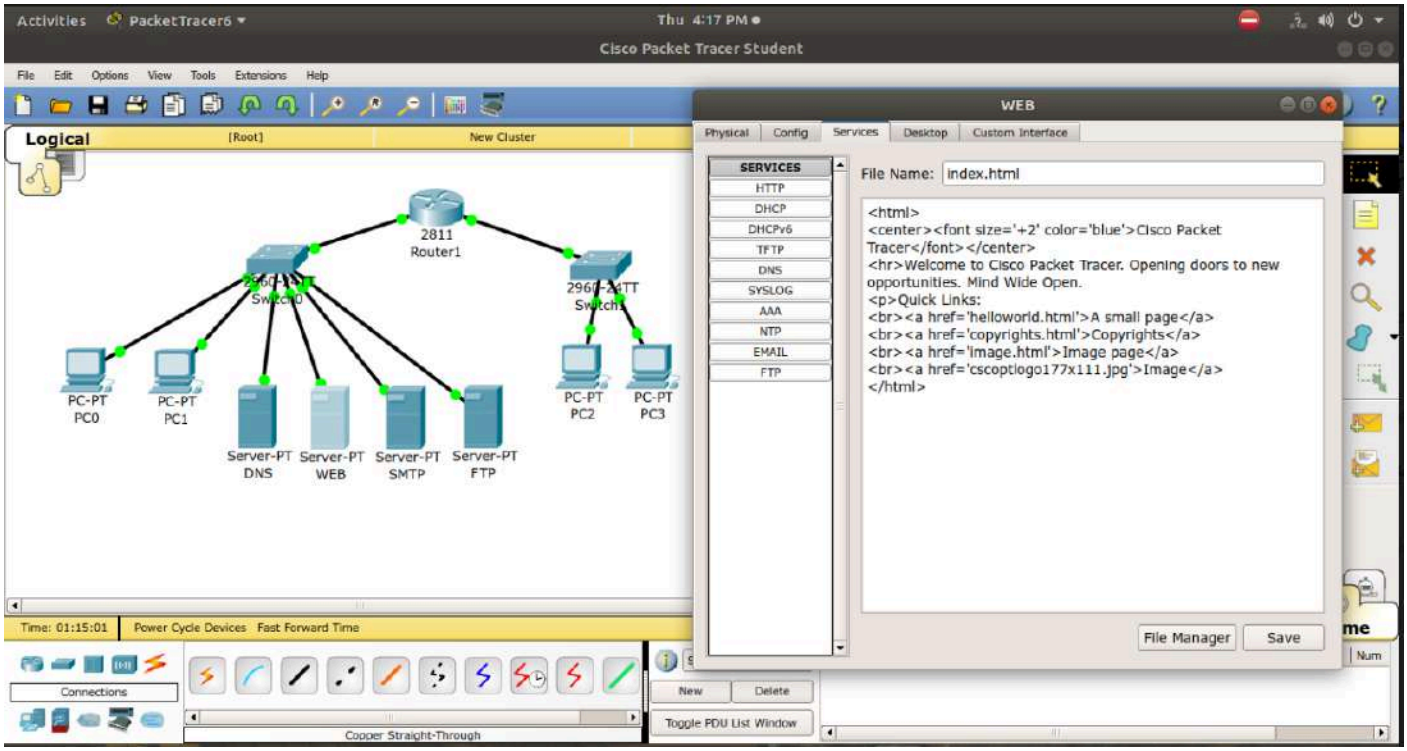
OUTPUT:

DNS SERVER-





WEB SERVER-

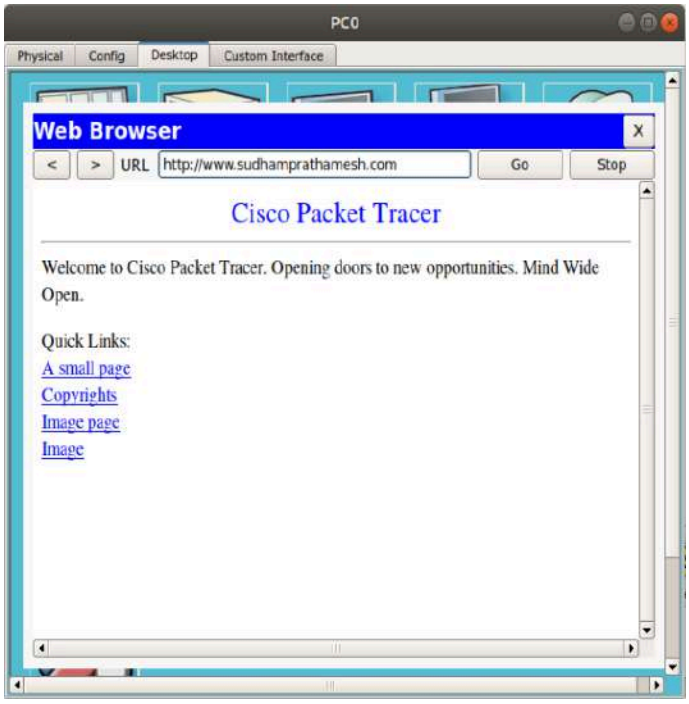

Cisco Packet Tracer Student
Logical
[Root]
New Cluster
2811
Router1
PC-PT
PC0
PC-PT
PC1
Server-PT
DNS
Server-PT
WEB
Server-PT
SMTP
Server-PT
FTP
PC-PT
PC2
PC-PT
PC3
WEB
Physical
Config
Services
Desktop
Custom Interface
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
File Name: index.html
<html>
<font size='+2' color='blue'>Cisco Packet Tracer</font>
<hr>Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.
<p>Quick Links:

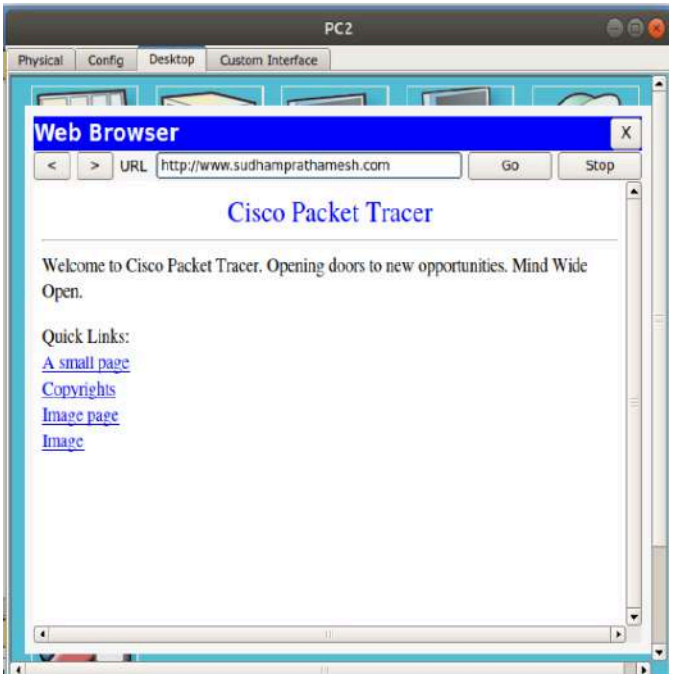
<a href='helloworld.html'>A small page</a>

<a href='copyrights.html'>Copyrights</a>

<a href='image.html'>Image page</a>

<a href='cscoptlogo177x111.jpg'>Image</a>
</html>
File Manager
Save
Time: 01:15:01
Power Cycle Devices
Fast Forward Time
Connections
Copper Straight-Through
New
Delete
Toggle PDU List Window


PC0
Physical
Config
Desktop
Custom Interface
Web Browser
URL
http://www.sudhamprathamesh.com
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


PC2
Physical
Config
Desktop
Custom Interface
Web Browser
URL
http://www.sudhamprathamesh.com
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

Name :- Prathamesh S. Chikankar

ROLL NO. :- AIML11 BRANCH :- CSE-(AI&ML)

YEAR :- TE SUBJECT :- WC (WEB COMPUTING)

Topic :- ASSIGNMENT NO. 01

Sign :- Prathu Date :- August'22

1) Write a short Note on HTTPS.

- HTTPS (Hyper text Transfer protocol secure) is a secure version of the HTTP protocol that uses the SSL/TLS protocol for encryption and authentication. HTTPS is specified by RFC 2818 (May 2000) & uses port 443 by default instead of HTTP's port of 80.
- HTTPS protocol make it possible for website users to transmit sensitive data such as credit card numbers, banking information, and login credentials securely over the internet. For this reason, HTTPS is especially important for securing online activities such as shopping, banking & remote work.
- This protocol uses a mechanism known as asymmetric public key infrastructure, and it uses two different keys which are given below:

1) private key: This key is available on the web server, which is managed by the owner of a website. It decrypts the information which is encrypted by public key.
2) public key: This key is available to everyone. It converts the data into an encrypted form.

The major difference b/w the HTTP and HTTPS is the SSL certificate. The HTTPS protocol is an extended version of HTTP protocol with an additional feature of security.

Advantages of https:

1) Secure communication: secure connection by establishing encrypted link b/w the browser and the server or any two systems.
2) Data Integrity: https provides data integrity by encrypting the data and so, even if hackers manage to hap the data, cannot read & modify it.
3) privacy and security: preventing hackers to passively listen to commn b/w browser and the server.
4) faster performance: increases speed of data transfer compared to http
5) SEO: use of https increases SEO ranking.
6) Future: https represents the future of the web by making internet safe for users & website owners.

2) Write a short note on Transport layer security.

- Transport layer security, or TLS, is a widely adopted security protocol designed to facilitate privacy and data security for communication over the Internet
- A primary use case of TLS in encrypting the communication b/w web application and servers, such as web browsing - loading a web site.
- TLS can also be used to encrypt other communication such as email, messaging and voice over IP (VoIP)
- defined in RFC 2246 - TLS is used by the http, smtp & ftp providing security transport layer.

derived from SSL

provides a secure searched connection b/w client & server (i.e. no third party)

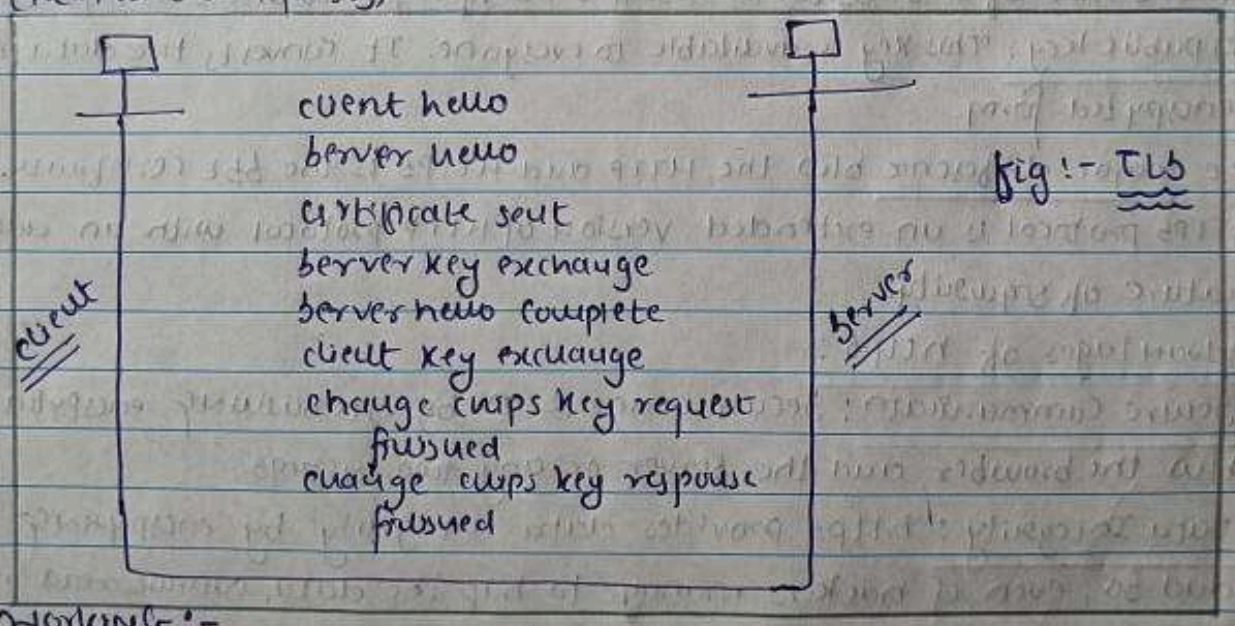


fig :- TLS

Working :-

1) Key exchange b/w client & server (by diffie Hellman algo)

2) Now TLS protocol will open an encrypted channel (by RC4/DSA/IDEA/AES algo.)

3) It also ensure that during the msg are not altered (by MD5/SHA algo.) - hashing algo.

* RFC2246 is similar to SSL V3.

3) What is DOM? Explain different types of DOM.

- Document object model (DOM) is a programming interface for HTML and XML (Extensible markup language) documents.
- It defines the logical structure of documents and the way a document is accessed and manipulated.
- The document object model (DOM) is a W3C standard. It defines a standard for accessing documents like HTML and XML.
- DOM is a way to represent the webpage in a structured hierarchical way so that it will become easier for programmers & users to glide through the document.
- DOM defines objects and properties and methods (interface) to access all XML elements. It is seperated into 3 different parts/levels:

1) Core DOM: Standard model for any structured document.
2) XML DOM: Standard model for XML document
3) HTML DOM: Standard model for HTML document

> XML DOM:

- XML DOM is a standard object model for XML, XML documents have a hierarchy of info. units called nodes; DOM is a standard programming interface of describing those nodes and the relationships b/w them.

Advantages: 1) XML DOM is language and platform independent.
2) XML DOM is traversable. 3) XML DOM is modifiable.

Disadvantages: 1) consumes more memory.
2) Operational speed, compared to SAX is slower.

> HTML: 1) Adds structure to our web pages
2) Tags used for eg. <P>

CSS: 1) Adds style to our web pages
2) eg:- colors, border, margin, image, position.

JS: 1) Add prog. to our web pages
2) Add funtionality, for eg:- client side validation, effects & events, etc..

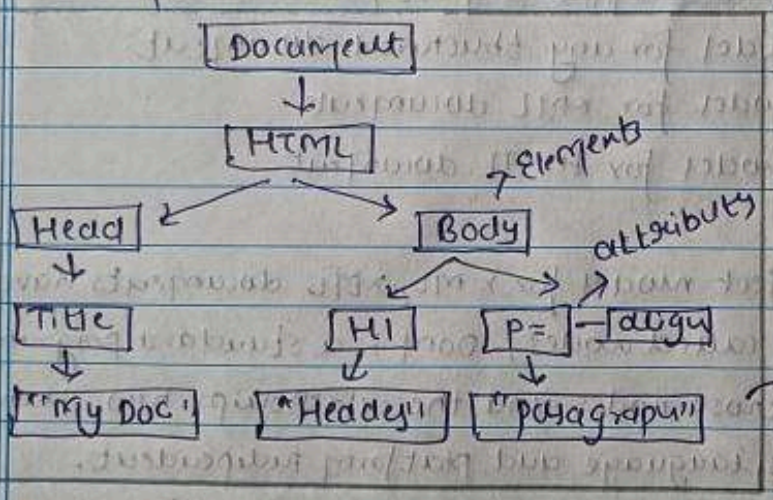
PTO

> DOM tree

A DOM tree is nodes represent on HTML or XML documents contents

eg:-
```
<html>
  <head>
    <title> my Document </title>
  </head>
  <body>
    <h1> Header </h1>
    <p> paragraph </p> <p align = "center">
  </body>
</html>
```

a DOM tree that looks like



⇒ when a web browser passes an HTML document, it builds DOM tree and then uses it to display mg document.

fig: DOM tree

4) write a short note on REST API.

- Representational State Transfer (REST) is a slw architectural style that defines the constraints to create web services. The web services that follow the REST architectural style is called REST full web services. It differentiate b/w computer System and web services.
- REST is a way of accessing web services in a simple and flexible way without having any processing.
- All communication done via REST API uses only HTTP Request.

PTO

> principles of REST API:

1. Uniform Interface: Defines the interface b/w client and server.
- Resource-based: the server cannot send the database but represents some database records expressed to HTML, XML or JSON depending on server request.
- manipulation of resources by representation
- self-descriptive message. (MIME type)
- As the engine of Hypermedia Application state.

2. client-server: interface separates the client from the server.
- for eg. the separation of concerns not having an internal relationship with internal storage for each server to improve probability of customer's data code

3. stateless: State of service doesn't persist b/w subsequent request and response. It means that the request itself contains the state required to handle the request. It can be a query-string parameters, entity, or header as a part of the URI.

4. layered and system: Directly connected to end server or by any intermediary whether a client cannot tell. Intermediate servers improve the system scalability by enabling load-balancing providing a shared cache. layers can enforce security policies.

5. cacheable: On the WWW, customers can cache responses. Therefore responses clearly define themselves as unacceptable or prevent customers from reusing stale or inappropriate data to further request.

6. Code on Demand: The server temporarily moves or optimizes the functionality of a client by logic that it executes. Eg: of compiled components are Java applets and client-side scripts.

> REST API methods:

→ features: - HTTP is a connectionless protocol.
- HTTP is stateless because both server & client keep track of last request.
- Any data can be transmitted & received via HTML HTTPS protocol, hence type-independent.

PTO

→ Standard HTTP methods are:
- GET: To fetch the related information.
- DELETE: To delete / for removing any record
- PUT: To update / upgrading information
- POST: To create and insert a new entry.

Also, HTTP methods in REST performing CRUD operation.

---

5. Explain the term URL and URI in detail.

→ - A URL is the address of a given unique resource on the web. each valid URL points to a unique resourse. Such as HTML page, a CSS document, an image etc.. URL → Uniform Resourse Locator...
- URL itself are handle by the web server
- URL eg. https://developer.mozilla.org
- URL is composed of different parts, some mandatory & other optional.

http://www.example.com:80/path/to/myfile.html?key1=value1&value2=value2#somewhere in the Document.

1) Scheme: http:// ⇒ which indicates that protocol that the behaviour must use to request the resources protocol is a set method for exchanging or transforming data around a computer n/w.

2) Authority: www.example.com:80
domain name, port

80 for HTTP & 443 for the HTPPS

3) path to resource: /path/to/myfile.html. ⇒ is the path to the resource on the web resource on the web server

4) parameter: ?key=1 value=1 & value2=value2.
Extra parameters provided to the web server. These parameters provided to the web server. These parameters are a list of key/value pairs separated with the ive & symbol.

5) Anchor: #somewhere In the Document

PTO

An anchor represents a sort of "bookmark" inside the resource, giveing the browser the position/direction to show content located at that "bookmarked" spot.

> How to use URLs

To create a links to other documents with the <a> element;
To link document with its related resources through various elements, such as <links> or <script>

Types: 1> Absolute URLs # 2> Relative URLs

⇒ - A URI or Uniform Resource Identifier is a STRING identifier that refers to a resource on the internet. It's a string of characters that is used to identify any resource on internet using location, name or both.

- A URI has two subsets; URL (Uniform resource locator) and URN (Uniform Resource Number). If it contains only a name, it means it's not a URL. Instead of directly URI, we mostly see the URL and URN in real world.

> Syntax of URI

Scheme : [//authority] path [?query] [#fragment]

1> Scheme: The first component of URI is scheme that contains sequence of characters that can be any combination of letters, digit, plus, sign, or hyphen (-), which is followed by colon (:). Popular schemes - http, file, ftp, data & rtc

2> Authority: optional and preceded by two slashes (//)
- uses info : @symbol. ; host : IP address []. ; port : optionals

3> path: path segment seperated by slash (/)

4> Query: optional, preceded by question mark (?) - non-hierarchical data.

5> fragment: (#) hash symbol. → direct to secondary source.

examples:

1. mailto : hey.Doe@example.com
2. news : comp.infosystems.www.servers.unix
3. urn. oasis : name : specification : docbook : dtd : rwe : 4.1.2

6) Explain built in objects in Java script with example.

- Built in objects are not related to any window or DOM object model.
- These objects are used for simple data processing in the JS.

⇒ 1) Math objects:

- Math object is a built-in static object
- It is used for performing complex math operation.

Math properties:

SQRT2 : returns square root of 2
PI : returns pi value 3.14
LOG2E : returns base 2 logarithm of E $\log_2 e$
LOG10E : returns base 10 logarithm of E $\log_{10} e$
LN10 : returns natural logarithm of 10 log 10

⇒ Maths methods:

abs(), acos(), log(), max(), min(), pow()

eg:-

```
<!DOCTYPE html>
<html>
<head>
    <title> JS get the absolute value of no. </title>
</head>
<body>
<script type="text/javascript">
    document.write (Math.abs(-1) + "<br>");
    document.write (Math.abs(5) + "<br>");
    document.write (Math.sqrt(4) + "<br>");
</script>
</body>
</html>
```

```
var value4 = Math.PI
document.write (" PI value is :" + value4 + "<br>");
```

PTO

Data objects:

- Data is datatype
- Data object manipulates date and time
- Data () constructor takes no arguments.
- Data object allows to get and set the year, month, day, hour, minute second and millisecond fields.

Syntax:

var variable-name = new Date();

eg:- var current-date = new Date();

Methods:

getDate(), getDay(), getFullYear(), Date(), getHours(), etc..

eg:-
```
<html>
<body>
<center>
    <h2> Date methods </h2>
    <script type = "text/javascript">
        var d = new Date();
        document.write("<br> Hours : <b>"
            + d.getHours() + "<br>");
//Returns the data objects as a string
        document.write("<b> Day: </b>" + d.getDay() + "<br>");
        document.write("<b>month: </b>" + d.getMonth() + "<br>");
</script>
</center>
</body>
</html>
```

String objects

- String object are used to work with text - a, b, c..

PTO

-It works with a series of characters

Syntax:

var variable_name = new String (string);

eg:- var s = new String (string);

Methods: charAt(), concat(), match(), includes()

```
<html>
<body>
<script type = "text/ Javascript">
var str = "web computing";
var s = str.split();
document.write("<br> charAt : <b>"+str.charAt(1)+"</br>");
document.write("<br> charCodeAt : <b>"+str.charCodeAt(3)+"<br>");
document.write("<br> lowerCase : <b>"+str.lowerCaseAt()+"<b></br>");
</script>
</body>
</html>
```

7) Explain window object in Javascript with example.

It is used to interact with the browser window which display the web page. It generally represents a tab information in browser like width and height will after the complete browser window.

-Open a new window

window.open (url, windowName, [windowFeatures])

Ex:- <!document html>

```
<head> <title> JS window obj property </title> </head>
<body> <h3> Create window </h3>
<button onclick = "Createwindow()" >
    Open a window </button>
<script>
```

```
function CreateWindow()
{ var win = window.open ("", "my window", "width" = 500
  *, * height = 200, screenx = 100, screeny = 100");
//window property
  var isClose = win.closed;
  var name = win.name;
//writing in the current window
  document.write (isClose + "<br>");
  document.write (name + "<br>");
  document.write (win.screeny + "<br>");
  document.write (win.screenx + "<br>");
  document.write ("Hellow world!!.");
}
```

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

→ window object represents a window in browser. An object of window is created automatically by the browser.

8) Explain different types of for loops in ES6 with example.

Looping statements in the programming languages help to execute the set of instructn / functn repeatedly while a condn evaluates to true

classificatn

- Loops
  - → Definite → for (;;)
    - ↳ for...in
  - → Indefinite
    - ↳ for...of
    - ↳ for (;;)
    - ↳ for...in

> The for (;;) loop: used to create iterate a part of program multiple times. If you have a fix no. of iteratn, then it is always recommended to use 'for' loop.

PTO

syntax:

```
for(initialization; condition; incr/decr){
  //statement or code to be executed
}
```

→ Initialization: executed once at starting of the loop.
→ condition: executed each time for testing condition of loop, until false.
→ Increment/decrement: increment/decrement the value
→ statement: represents body of the loop, executed every time, until false

example: table of 2 using for loop.

```
var i;
for (i=1; i<10; i++)
{
   console.log("2*x" + i + "=", 2*i);
}
```

2 x

> The for...in loop: similar to for loop, which iterates through the property of an object, i.e., when you require to visit the properties or keys of object, then you can use in for...in loop.

syntax:

```
for(variable_name in object_name) //here in is the keyword
{
  //statement or block to execute
}
```

Every iteration, one property from object is assigned to name of variable, and this loop continues till all of the object properties get covered.

example: iterate through an object.

```
const student ={
    name: 'Manu',
    class : TE,
    age : 20
}
```

PTO

```
//using for...in
for (let key in student){
    //display the properties
    console.log(`${key} => ${student[key]}`);
}
```

In above prog. for..in loop is used to iterate over the student object
- object key is assigned to variable key.
- student[key] is used to access the value of key

> The for..of loop

unlike object iterators, this loop is used to iterate the iterable (array, etc..)

Syntax:

```
for (variable_name_of object_name) //here of is the keyword
{
    //statement or block of execution
}
```

In every iteration, one property from the iterable gets assigned to variable_name and the loop continues till end of iteration.

example:

```
var fruits = ["Apple","Banana","Mango"];
for (let value of fruits){
  console.log(value);
}
```

---

9) Explain browser object model in java script.

following are some important browser objects provided by Java Script

> window object: Represents window in browser.

object of window is created automatically by browser.

Methods of window object:

- alert() => displays the alert box containing message with OK button.

PTO

- confirm() => displays confirm dialog box containing message with ok and cancle button.
- open() => opens new window
- close() => close current window
- etc..

> Navigator object: used for browser detection. It can be used to get browser information such as appName, appCodeName, userAgent etc.. window property, so it can be accessed by
window.navigator OR navigator

methods:
- javaEnabled() => checks if java is enabled
- taintEnabled() => checks if taint is enabled.

example:
- navigator.appName
- navigator.appCodeName
- navigator.platform
- navigator.appVersion

> History object: represents an array of URLs visited by the USER. By using this object, you can load previous, forward or any particular page. Accessed by,
window.history OR history

methods:
- forward() => loads next page
- back() => loads previous page
- go() => loads given page no.

eg:-
```
<body> <h3> History length property </h3>
<script>
        //fetch no. of visited pages
document.write("you visited" + window.history.length
                + " pages.")
</script> </body>
```

PTO

> location object: Helps in storing the info. of current URL of window object.
child object of window object.

methods:
- assign() ⇒ loads new document in browser
- reload() ⇒ reloads current document contained in location.href property
- replace() ⇒ replaces current document with specified new one.

example:

```
<script>
    document.write(location.host);
</script>
```

properties: host, protocol, href, port, pathName, etc..

eg:- www.javascript.com or IP address

> document object: represent HTML document loaded in browser.
provides access to all HTML elements of document (web pages)

example:

```
document.write(document.domain + "<br>");
document.write(document.title + "<br>");
document.write(document.lastModified + "<br>");
```

> screen object: Built in object used to fetch information related to the browser screen like screen size etc.. Also obtained from window object.

eg:-

```
document.write("Total Height" + screen.height + "<b>");
document.write("Total width" + screen.width + "<b>");
document.write("Screen Colour Depth" + screen.colorDepth + "<b>");
```

property:

availability & available width: specifies the width and height of screen, excluding the window taskbar

width & height - specifies total width height of screen

10) write difference b/w ES5 and ES6.

parameter	ES5	ES6
defn:	ES5 is the fifth edition of the ECMAScript	ES6 is the sixth edition of the ECMAScript
Release:	2009	2015
Data-type:	Supports primitive data type that are string number, boolean, null & undefined.	There are some addition to Javascript data types. 'Symbol'.
Defining variable	Using a var keyword	Using let and const
performance	prior → non-presence of some features, so it has a lower performance than ES6	Because of new features & the shorthand storage implementation ES6 has high performance.
Support	wide range of community support it	Has lot of ~~communication~~ support but lesser than ES5 community
Object manipulation	ES5 is time-consuming than ES6	due to destructuring & speed operators, manipulation mox smoothly in ES6.
Arrow function	In ES5, both function & return keywords are used to define a function	arrow function → new feature introduced in ES6 by which we didn't require the function.
Loops	use of for loop to iterate over elements	introduce concept of for...of loop to perform iteration over value of iterable objects.

LTCE

AIML11
pg. no. a)

Name :- Prathamesh S. Chikankar
ROLL NO :- AIML11 Branch :- CSE-(AI&ML)

Year :- TE Subject :- Web Computing (Computer N/w)
Topic :- Assignment No. 02

Date of submission :- October'22 Sign :- Prathamesh

1. Difference Between

	TCP	UDP
1	TCP stands for Transmission control protocol	UDP stands for User Datagram protocol
2	TCP is a connection-oriented protocol.	UDP is connection less protocol
3	TCP is highly reliable	UDP is suitable efficient transmission
4	TCP is used by HTTP, HTTPs, SMTP, Telnet	UDP is used by DNS, DHCP, SNMP VOIP.
5	ordering of packet is maintained by TCP	UDP has packets that is independent of each other.
6	speed is slower than UDP	speed is fasty than TCP
7	Header size is 20 bytes	Header size is of 8 bytes
8	It has Ack segment	It has not Ack.

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 N/w with a shared transmission medium.

3) The extension also regulates how to proceed if collision occur ie, when two or more nodes try to send data packets via the transmission medium [bus] simultaneously & they interface each other

4) To understand CSMA/CD works, we break down the components:-

i) carrier sense (CS): The carrier state detection make sure that all N/w participants check whether the medium is current free; only then does the protocol initiate the data transmission.

ii) multiple access (MA): several participants [computers connected

PTO

to the N/w] 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:

① CSMA/CD is technique used for shared medium access
② CSMA/CD is used in Ethernet.
③ It detect collision with in a short time
④ It avoids wastefull transmission.

3. Draw Header format of TCP datagram.

20-60 Byte → Header size

Header	Data

Source port No.	Destination port No.
16 bit	16 bit
Sequence no. 32 bit	
ACK no. 32 bit	
HLEN 4 bit / Reserve 6 bit / flags 6 bit	window size 16 bit
checksum 16 bit	urgent pointer 16 bit
option & padding	

fig:- Header format of TCP

Flags: URG, ACK, PSH, FIN, SYN, RST

4. Explain different N/w layer function:

Duties of N/w layer

Internetworking, Addressing, Routing, Parketizing, fragmenting

i) Internetworking: This is the main duty of N/w layer. It provides the logical connection b/w different layers types of network.

ii) Addressing: Addressing is necessary to identify each device on the

PTO

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 N/w, there are multiple routes available from a source to destination and one of them is to be choosen.
The N/w layer decides the root to be taken. This is called as routing and it depends on various criterians.

iv) Packetizing: As discussed earlier, the N/w layer encapsulates the packets recived 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 recieved frame. Then the datagram is processed and encapsulated in another frame.

5. An ISP granted a block of addressing with 150.80.0.0/16. The ISP wants to distribute these blocks to 2600 customers follows:
a) 1st group has 200 medium size businesses & each needs 128.
b) 2nd group has 400 small businesses & each needs 16.
c) 3rd group has 2000 households each needs 4.
Design the suitable subblock of given slash notation for each subblock. and find out the remaining addresses that are available.

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

- Group 1: $128 = 2^{7} \rightarrow a$ $\quad 32-a = 32-7 = 25 = n$, subnet

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

Business ②: 150.80.0.128/25
150.80.0.255/25
⋮
Business (200): 150.80.99.128/25
150.80.99.255/25

Total addresses
$\Rightarrow 200 \times 128 \Rightarrow 25600$

PTO

Group 2:- $16 = 2^{4} \rightarrow a = 32-a \Rightarrow 32-4 = 28 \Rightarrow n_2$ 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

$\frac{256}{16}$ 16 for 400th BM
$400/16 \Rightarrow 25$

Total address = 400 × 16
= 6400

Group 3: $4 = 2^{2} \rightarrow a = 32-a = 32-2 \Rightarrow 30 \Rightarrow n_3$ 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 (1984) : 150.80.155.252/30
150.80.155.255/30

⋮

Business (2000) : 150.80.156.60/30
150.80.156.63/30

$\frac{256}{4} = 64$ $\frac{2000}{64}$ 31 25
31 × 64
= 1984/64 31/

126 + 31 = 157
125 + 31 = 156

Total address = 2000 × 4
= 8000

∴ Total remaining addresses:
65536 − 40000 ⇒ 25536 //

6. Explain IPv4 header format & draw:

> VERSION : Version of internet protocol used (eg:- IPv4) – 4bits

> HLEN (4 bit) : Header length ; length of entire IP header.

> Service (8 bit) : Type of service

Header				Data
VER	HLEN	Service	Total length	
4 bit	4 bit	8 bit	16 bit	
Identification field			flags	fragmn
(16 bit)			3 bit	13 bit
TTL (8 bit)	protocol (8 bit)	Header checksum (16 bit)		
Source IP address 32 bit				
Destination IP address 32 bit				
Option (If HLEN) is >32 bit				

fig:- IPv4 header format

> Total length : length of entire packet {Header + data} (16 bit)
> Identification : used in fragmentation (16 bit)
> flags : can be fragmented or Not (MSB) set to '0' (3 bits)
> fragment offset : Extra position of fragment in original IP packet (13 bit)
> Time to leave (TTL) : to avoid looping (8 bits)
> protocol : to which protocol packets belongs to (8 bit)
> Header checksum : packet recieved error free or Not (16 bit)
> source address : sender (32 bit)
> destination address : receiver (32 bit)
> option : sequencing (32 bit)

7. Write a short Note on NAT

① NAT: Network addressing Translation was designed by 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 no. of uniquely public IP addresses.

② NAT is generally used in different types of corporation, where a NAT router is placed at the edge of the private network and transform the

PTO

IP address which are attached to packet

③ The NAT routes, which performs as an effective agent between the InternNet and the local network.

④ For eg, when a computer is placed in a private N/w it request data from the public N/w.

⑤ When the information is been requested is returned, the NAT routes checks its internal list of public to private address mappings for the purpose of forwarding the information to the suitable computer.

> Types of NAT : i. Static NAT environment
ii. Dynamic NAT environment

i) Static NAT environment:

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

ii) Dynamic NAT environment:

In a dynamic NAT environment, the NAT routes dynamically allocate public IP addresses, from a group of addresses, to devices on private N/w that wish to communicate with the public N/w.

8. Difference B/w

	IPv4	IPv6
1	IPv4 has 32 bit add. length	IPv6 has 128 bit address length
2	It has support for manual and the DHCP configuration	It has support for auto-configuration and renumbering.
3	Availability of max 4.3 billion addresses	Has limitless no. of IP addresses
4	Security is dependent on application	for security IPSEC is inbuilt
5	optional field is available	optional field isn't available
6	Broadcasting message transmission	Multicasting message transmission
7	Address represen in decimal form	Address represn in Hexadecimal form

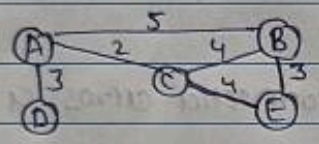
9. Short Note on Dijkstra Algorithm.

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 tintative or permanent. Initially all labels are tintative.

④ when it discovered that node, it is made permanent and more never changed the shortest path possible path from source to that node, N/w can have more than one shortest path.

eg:- 

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)	–

10. Explain CISCO service oriented N/w Application

① The CISCO service oriented N/w architecture (SONA) approach provides a standard provider paradism for designing Networks that link N/w services with application to drive business value by:

a. Establishing rapid adoption & n/w deployement of new appⁿ services.

b. co-ordinating application & N/w events with business procedure.

c. Aligning N/w resources to application to meet business objectives objective

PTO

② It is a funcm framework that enables business to build an intelligent entreprise-wide infrastructure. SONA accomplishes this by separating the N/w architecture.

> Benefits of Cisco SONA:

① functionality: supports the organizational requirements
② Scalability: supports growth & expansion of organizational task.
③ Availability: provide necessary services reliable anywhere, anytime
④ performance: provide desired responsiveness, throughputs & utilizatn.
⑤ manageability: provides control, performance monitoring & fault detectn.
⑥ Efficiency: SONA provides N/w services & info. with reasonable cost.

> There are 3 layers of Cisco SCAN. they are:

① The N/w infrastructure layer:
- where all the IT resources are interconnected across a converged N/w fundatn.
- The objective in these layers is to be have connectivity with customers anytime & anywhere.

② The interactive services layer:
- It enables efficient allocatn of resources to applicatn & business processes delivered through the N/w infrastructure.
- The integrated N/w service layer hold key N/w infrastructure services, including security, mobility, storage, unified communicatn, computing, applicatn delivery & identity.

③ The applicatn layer:
- It contains the business applicatn and collaborative applicatn that are take advantage of efficiency from the interactive services.
- The objective for customers in this layer is to meet business requirements and achieve efficiencies by leveraging the interactive services layer.