

Internet Technologies

1. Display your systems IP Address, Subnet mask using ipconfig, and find out the network address and the maximum number of systems possible on your network and range of IP addresses available to these systems.

ipconfig for:

- **System IP Address:** 192.168.43.165
- **Subnet mask:** 255.255.255.0
- **Network ID:** 192.168.43.0
- **Broadcast ID:** 192.168.43.255
- **Maximum number of system possible on the network:** 254
- **Range of IP addresses:** 192.168.43.0 to 192.168.43.255
- **Class type:** C

2. With help of ping, check if you are connected to other systems of your network and find the route to connect to that system using tracert. List all the processes which are using ports for TCP protocol.

ping google.com check if you are connected to other systems of your network

tracert google.com find the route to connect to that system

List all the processes which are using ports for TCP protocol

netstat -r for route table

netstat -a for active connections

3. Create an HTML page that shows information about you, your course, hobbies, address, and your plans. Use CSS for styling of HTML page so that it looks nice.

- Make 2 files index.html and main.css (link main.css in index.html)

```
<link rel="stylesheet" href="./main.css" />
```

- Give heading and your name inside a header and give them styling
- Write about me section, hobbies, learnings using and tags and add your socials.

```
.content-footer,  
.tophead-heading,  
.tophead-intro {  
    text-align: center;  
}
```

4. Create an HTML page with the sole purpose to show multiplication tables of 2 to 10 (row-wise) created by JavaScript. Initially, the page is blank. With the help of setInterval function, print a row every 5 seconds in different colors and increasing font size.

```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
    <title>Multiplication Table</title>  
    <link rel="stylesheet" href="./css/style.css">  
</head>  
  
<body>  
    <h1>Multiplication table</h1>  
    <main id="main"></main>  
    <script src="./js/script.js"></script>  
</body>  
  
</html>
```

// script.js

```
const mainEl = document.getElementById('main');

var i = 2;
var j = 1;
var t = 1;

var colors = ['#002B5B', '#AAC4FF', '#8BBCCC', '#006778', '#FF5677'];

const childhead = document.createElement('h3');
childhead.innerHTML = `Table of ${i}`;
mainEl.appendChild(childhead);
setInterval(() => {
    if (j > 10) {
        i++;
        j = 1;
        const childhead = document.createElement('h3');
        childhead.innerHTML = `Table of ${i}`;
        childhead.style.color = colors[(j - 1) % 5];
        childhead.style.fontSize = `${t}rem`;
        mainEl.appendChild(childhead);
    }

    if (i > 10) return;

    const child = document.createElement('h5');
    child.innerHTML = `${i} * ${j} = ${i * j}`;
    child.style.color = colors[(j - 1) % 5];
    child.style.fontSize = `${t}rem`;
    mainEl.appendChild(child);
    window.scrollTo(0, document.body.scrollHeight);

    j++;
    t += 0.05;
}, 5000);
```

5. Create an HTML page with a paragraph written on it and under which 9 buttons are placed in a 3X3 grid. The first row is for buttons labeled with colors names Red, Green, and Blue, the second row with numbers 10, 20, 30, and the third row with different font names. Click event of each of the buttons should make the appropriate change in the style of paragraph.

Inside index.html write a paragraph in <p> tag and make grid

```
<main class="grid">
    <button id="c-red">Red</button>
    <button id="c-green">Green</button>
    <button id="c-blue">Blue</button>
    <button id="s-10">10</button>
    <button id="s-20">20</button>
    <button id="s-30">30</button>
    <button id="f-ss">Sans Serif</button>
    <button id="f-s">Serif</button>
    <button id="f-m">Monospace</button>
</main>
```

And then call script

```
<script src="./script.js"></script>
```

```
const para = document.querySelector('p');

document.getElementById('c-red').addEventListener('click', () => {
    para.style.color = 'red';
});

document.getElementById('c-green').addEventListener('click', () => {
    para.style.color = 'green';
});

document.getElementById('c-blue').addEventListener('click', () => {
    para.style.color = 'blue';
});

document.getElementById('s-10').addEventListener('click', () => {
    para.style.fontSize = `${10}px`;
});

document.getElementById('s-20').addEventListener('click', () => {
    para.style.fontSize = `${20}px`;
});
```

```
document.getElementById('s-30').addEventListener('click', () => {  
    para.style.fontSize = `${30}px`;  
});  
  
document.getElementById('f-ss').addEventListener('click', () => {  
    para.style.fontFamily = 'sans-serif';  
});  
  
document.getElementById('f-s').addEventListener('click', () => {  
    para.style.fontFamily = 'serif';  
});  
  
document.getElementById('f-m').addEventListener('click', () => {  
    para.style.fontFamily = 'monospace';  
})
```

6. Create a form that takes data about a pet. The form must be well designed and should accept the pet's name, age, weight, type, and what it likes most. At the submission of this form create a Pet object in JavaScript filled with these values and log that object and equivalent JSON on the console.

```
<body>
  <div class="container">
    <h1>Pet's Information</h1>
    <hr>
    <label for="name">Pet's Name: </label>
    <input type="text" name="name"><br><br>
    <label for="age">Age: </label>
    <input type="number" name="age">
    <label for="weight">Weight: </label>
    <input type="number" name="weight" class=""><br><br>
    <label for="type">Pet type: </label>
    <input type="text" name="type"><br><br>
    <label for="likes">Likes: </label>
    <input type="text" name="likes"><br>
    <button class="btn-submit" onclick="display()">Submit</button>
  </div>

  <script type="text/javascript">

    function display(){
      // event.preventDefault();
      var pet = {};
      var input_fields =
document.getElementsByTagName('input');
      for (var i = 0; i < input_fields.length; i++) {
        pet[input_fields[i].name] = input_fields[i].value;
      }
      console.log(pet);
    }

  </script>

</body>
```

7. Store JSON data of a few pets that you created in previous practical in a JSON file (copy from console output of previous program to a .json file). Using AJAX, load data from the file and display it in a presentable way using HTML and CSS.

8. Create a plain HTML page for B.Sc. Hons CS course, mentioning details like fee, eligibility criteria, papers with names and credits, and future possibilities after the course. A button for styling should be there at the bottom of the page. On clicking on this button JavaScript should redesign the complete page using jQuery in a nice presentable way.

9. Create an HTML page for an image gallery which shows the use of BOOTSTRAP to rearrange and resize its contents on resizing the browser.

```
// index.html
```

[illegible]

```
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></sc
ript>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"><
/script>

</body>
</html>
```


10. Create an HTTP server using Node.js which handles requests on port 10000 or a free port beyond 10000. Modify the server in such a way that opening localhost:10000 will display “Hello world, This is my Node.js server” on the browser.

npm init

// creating nodejs server index.js

```
const projectName = "Node.js-Hello-World";
const http = require("http");

//create a hostname and port number
const hostName = "127.0.0.1";
const port = 10000;

//create server
const server = http.createServer((request, result) => {
  result.statusCode = 200;
  result.setHeader("Content-Type", "text/plain");
  result.end("Hello World, this is my Node.js server");
});

//set server to listen
server.listen(port, hostName, () => {
  console.log(`Server running at http://${hostName}:${port}/`);
});
```

node index.js

Server will start

11. Create index.html file containing two forms for SignIn and SignUp.

- Submitting a SignIn form should search for credentials in the mysql database using a server created in previous practical. On successful sign in, a welcome page should be displayed.
- Submitting the SignUp form should insert a new entry for credentials in the mysql database using the server created in previous practical. On successful sign up, user should be returned back to index.html

QUESTION: Create an HTML form in a file called car.html to get the following details of the car:

Name of the manufacturer

Name of the model

Manufacturing Year

Fuel type (petrol/diesel)

Color

Seating capacity

Cubic capacity

Write JQuery code in a JavaScript file car.js to get the details of the car from the car.html on pressing submit button. Also write the JavaScript code in car.js to make a JavaScript object and JSON object from the above details and print both the JavaScript and JSON object on the console

// car.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cars</title>
  <link rel="stylesheet" href="car.css">
</head>
<body>
  <form>
    Name of the manufacturer: <input type="text"
id="manufacturer"/><br>
    Name of the model: <input type="text" id="model"/><br>
    Manufacturing Year: <input type="text" id="mYear"/><br>
    Fuel Type (petrol/diesel): <input type="text" id="fType"/><br>
    Color: <input type="text" id="color"/><br>
    Seating Capacity: <input type="text" id="sCapacity"/><br>
    Cubic Capacity: <input type="text" id="cCapacity"/><br>
    <input type="submit" id="submit">
  </form>

  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></sc
ript>
  <script src="car.js"></script>

</body>
</html>
```

// car.js

```
$(function(){
    $form = $('#form');
    $form.on('submit', function(e){
        e.preventDefault();
        var manufacturer = $('#manufacturer').val();
        var model = $('#model').val();
        var manYear = $('#mYear').val();
        manYear = Number(manYear);
        var fuelType = $('#fType').val();
        var color = $('#color').val();
        var seatingCapacity = $('#sCapacity').val();
        var cubicCapacity = $('#cCapacity').val();
        cubicCapacity = Number(cubicCapacity);

        const car = {
            cManufacturer: manufacturer,
            cModel: model,
            cManfYear: manYear,
            cfuelType: fuelType,
            cColor: color,
            cSeatingCapacity: seatingCapacity,
            cCubicCapacity: cubicCapacity
        };

        // Json object
        var jCar = JSON.stringify(car);
        console.log('JSON object is: ', jCar);

        // Javascript object
        var carJS = JSON.parse(jCar);
        console.log('Javascript object is: ', carJS);
    });
})
```