Advance JavaScript

Module 1 (Introduction and Code Quality) :-

1). Write a program to Show an alert.

```
<!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>Alert</title>
  </head>
  <body>
    <h2>The alert() Method</h2>

    <p>Click the button to see line-breaks in an alert box.</p>

    <button onclick="myFunction()">Try it</button>

    <script>
      function myFunction() {
        alert("Hello\n My name is jaymin");
      }
    </script>
  </body>
</html>
```

2). What will be the result for these expressions?

1). 5 > 4 :-

```
<!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>Document</title>
</head>
<body>
    <script>
      var x = 5;
      var y = 4;
```

```
      if ( x > y )
      {
        document.write("True");
      }
      else
      {
        document.write("False");
      }

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

Ans. True

2). “apple” > “pineapple” :-

```
<!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>Document</title>
</head>
<body>
    <script>
      var x = "apple";
      var y = "pineapple";

      if ( x > y )
      {
        document.write("True");
      }
      else
      {
        document.write("False");
      }

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

Ans. False

3). “2” > “12” :-

```
<!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>Document</title>
</head>
<body>
    <script>
      var x = "2";
      var y = "12";

      if ( x > y )
      {
        document.write("True");
      }
      else
      {
        document.write("False");
      }

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

Ans. True

4). Undefined == null :-

```
<!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>Document</title>
  </head>
  <body>
    <script>
      var x;
      var y = null;

      if (x == y) {
        document.write("True");
```

```
      } else {
        document.write("False");
      }
    </script>
  </body>
</html>
```

Ans. True

5). Undefined === null :-

```
<!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>Document</title>
  </head>
  <body>
    <script>
      var x;
      var y = null;

      if (x === y) {
        document.write("True");
      } else {
        document.write("False");
      }
    </script>
  </body>
</html>
```

Ans. False

6). Null == “\n0\n”

```
<!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>Document</title>
  </head>
  <body>
    <script>
      var x=null;
```

```
      var y = "\n0\n";

      if (x == y) {
        document.write("True");
      } else {
        document.write("False");
      }
    </script>
  </body>
</html>
```

Ans. False

7). Null === + “\n0\n”

```
<!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>Document</title>
  </head>
  <body>
    <script>
      var x=null;
      var y = "\n0\n";

      if (x === + y) {
        document.write("True");
      } else {
        document.write("False");
      }
    </script>
  </body>
</html>
```

Ans. False

3). Will alert be shown?

if ("0") { alert('Hello'); }

```
<!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>Document</title>
  </head>
  <body>
    <script>
      if ("0") { alert( 'Hello'); }
    </script>
  </body>
</html>
```

Ans.

4). What is the code below going output ?

Ans.

```
<!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>Document</title>
  </head>
  <body>
    <script>
      alert( null || 2 || undefined );
    </script>
  </body>
</html>
```

Output :-

5). The following function returns true if the parameter age is greater than 18. Otherwise it asks for a confirmation and returns its result.

```
<!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>Document</title>
  </head>
  <body>
    <script>
      const checkAge = 12;
      if (checkAge>18)
      {
        document.write("true");
      }
      else
      {
        confirm("did parents allow you?");
      }
    </script>
  </body>
</html>
```

6). Replace function expression with arrow functions in the code below.

Ans.

```
<!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>Document</title>
  </head>
  <body>
    <script>
      var ask = (question, yes, no) => {
        if (comfirm(question)) {
          yes();
        } else no();
      };

      // var ask = ("Do You Agree?", ()=>{alert("You Agreed.")})=>{
      //   alert("alert");
      // };
    </script>
  </body>
</html>
```

Module 2 (Data Types and Objects) :-

1). Write the code, one line for each action:

A). Create an empty object user :-

```
<!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>Document</title>
  </head>
  <body>
    <script>
      var objectA={};
      var objectB=new Object();
      </script>
  </body>
</html>
```

B). Add the property name with the value John :-

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
const person = {
  firstname: "John"
};

document.getElementById("demo").innerHTML = person.firstname;
</script>

</body>
</html>
```

C). Add the property surname with the value Smith :-

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
const person =
{
  surname: "Smith"
};

document.getElementById("demo").innerHTML = person.surname;
</script>

</body>
</html>
```

D). Change the value of the name to Pete. :-

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>
```

```
<script>
const person = {
  firstname: "Pete"
};

document.getElementById("demo").innerHTML = person.firstname;
</script>

</body>
</html>
```

E). Remove the property name from the object. :-

```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
var person = {
  firstname: "John",
  surname: "Smith"
};

delete person.surname;
document.getElementById("demo").innerHTML =
person.firstname + " and surname is " + person.surname;
</script>

</body>
</html>
```

2). Is array copied?

```
<!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>Document</title>
  </head>
```

```
  <body>
    <script>
      "use strict";

      let fruits = ["Apples", "Pear", "Orange"];

      let shoppingCart = fruits;

      shoppingCart.push("Banana");
      document.write(shoppingCart);
      alert(fruits.length); // 4
    </script>
  </body>
</html>
```

3). Map to names?

```
<!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>Document</title>
</head>
<body>
  <script>


    let john = { name: "John", age: 25 };
    let pete = { name: "Pete", age: 30 };
    let mary = { name: "Mary", age: 28 };

    let users = [ john, pete, mary ];

    let names = users.map(item => item.name);
    document.write(names);
    alert( names ); // John, Pete, Mary
    </script>
</body>
</html>
```

4). Map to objects

```
<!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>Document</title>
</head>
<body>
  <script>


    let john = { name: "John", surname: "Smith", id: 1 };
    let pete = { name: "Pete", surname: "Hunt", id: 2 };
    let mary = { name: "Mary", surname: "Key", id: 3 };

    let users = [ john, pete, mary ];

    let usersMapped = users.map(user => ({
      fullName: `${user.name} ${user.surname}`,
      id: user.id
    }));

    /*
    usersMapped = [
      { fullName: "John Smith", id: 1 },
      { fullName: "Pete Hunt", id: 2 },
      { fullName: "Mary Key", id: 3 }
    ]
    */

    alert( usersMapped[0].id ); // 1
    alert( usersMapped[0].fullName ); // John Smith
    </script>
</body>
</html>
```

5). Sum the properties There is a salaries object with arbitrary number of salaries. Write the function sumSalaries(salaries) that returns the sum of all salaries using Object.values and the for..of loop.If salaries is empty, then the result must be 0.

```
<!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>Document</title>
  </head>
  <body>
    <script>

      function sumSalaries(salaries) {
        let sum = 0;
        for (let salary of Object.values(salaries)) {
          sum += salary;
        }

        return sum; // 650
      }

      let salaries = {
        John: 100,
        Pete: 300,
        Mary: 250,
      };

      alert(sumSalaries(salaries)); // 650
    </script>
  </body>
</html>
```

6). Destructuring assignment We have an object: Write the Destructuring assignment that reads:

A) Name property into the variable name.

```
<!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>Document</title>
  </head>
  <body>
    <script>

      let user = {
        name: "John",
        years: 30,
      };
```

```
      let { name, years: age, isAdmin = false } = user;

      alert(name); // John
      alert(age); // 30
      alert(isAdmin); // false
    </script>
  </body>
</html>
```

B) Year's property into the variable age.

```
<!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>Document</title>
  </head>
  <body>
    <script>

      let user = {
        name: "John",
        years: 30,
      };

      let { name, years: age, isAdmin = false } = user;

      alert(name); // John
      alert(age); // 30
      alert(isAdmin); // false
    </script>
  </body>
</html>
```

C). isAdmin property into the variable isAdmin (false, if no such property) :-

```
<!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>Document</title>
  </head>
```

```
  <body>
    <script>

      let user = {
        name: "John",
        years: 30,
      };

      let { name, years: age, isAdmin = false } = user;

      alert(name); // John
      alert(age); // 30
      alert(isAdmin); // false
    </script>
  </body>
</html>
```

D) let user = { name: "John", years: 30}; :-

```
<!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>Document</title>
  </head>
  <body>
    <script>

      let user = {
        name: "John",
        years: 30,
      };

      let { name, years: age, isAdmin = false } = user;

      alert(name); // John
      alert(age); // 30
      alert(isAdmin); // false
    </script>
  </body>
</html>
```

7). Turn the object into JSON and back Turn the user into JSON and then read it back into another variable.

user = { name: "John Smith", age: 35};

```
<!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>Document</title>
  </head>
  <body>
    <script>
      let user = {
        name: "John Smith",
        age: 35,
      };

      let user2 = JSON.parse(JSON.stringify(user));
    </script>
  </body>
</html>
```

Module 3 (Document, Event and Controls) :-

1). Create a program to Hide/Show the password :-

```
<!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>Document</title>
  </head>
  <body>
    <b><p>Click on the checkbox to show or hide password:</p></b>

    <b>Password</b>:
    <input type="Password" id="Typepass" />

    <input type="checkbox" onclick="Toggle()" />
    <b>Show Password</b>
```

```
    <script>
      // Change the type of input to password or text
      function Toggle() {
        var temp = document.getElementById("Typepass");
        if (temp.type === "Password") {
          temp.type = "text";
        } else {
          temp.type = "Password";
        }
      }
    </script>
  </body>
</html>
```

2). Create a program that will select all the classes and loop over and whenever i click the button the alert should show :-

```
<!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>Document</title>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.1/jquery.min.js"></scrip
t>
  </head>

  <body style="text-align: center">

    Username:<input id="uname" type="text" /><br />
    Password:<input id="pass" type="password" /><br />
    <button id="GFG_Button">Submit</button>

    <script>
      $("#pass").keypress(function (event) {
        if (event.keyCode === 13) {
          $("#GFG_Button").click();
        }
      });

      $("#GFG_Button").click(function () {
        alert("Button clicked");
```

```
      });
    </script>
  </body>
</html>
```

3). Create a responsive header using proper JavaScript :-

HTML code :-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="style.css"/>
    <title> Responsive Navigation Bar </title>
</head>
<body>
    <nav>
        <div class="logo">
            <!-- <img decoding="async" src="" alt="Logo Image"> -->
        </div>
        <div class="hamburger">
            <div class="bars1"></div>
            <div class="bars2"></div>
            <div class="bars3"></div>
        </div>
        <ul class="nav-links">
            <li><a href="#">HTML & CSS</a></li>
            <li><a href="#">WordPress</a></li>
            <li><a href="#">Javascript</a></li>
            <li><a href="#">JQuery</a></li>
            <li><a href="#">Contact Us</a></li>
            <li><button class="login-button" href="#">Sign In</button></li>
        </ul>
    </nav>
    <script src="script.js"></script>
</body>
</html>
```

CSS code :-

```
*{
    margin:0; padding:0;
```

```
    color:#f2f5f7;
    font-family: sans-serif;
    letter-spacing: 1px;
    font-weight: 300;
}

body{
    overflow: hidden;

}

nav{
    height: 6rem;
    width: 100vw;
    display: flex;
    position: fixed;
    z-index: 10;
    background-color: #053742;
    box-shadow: 0 3px 20px rgba(0,0,0,0.2);
}

/* Styling Navigation Links*/

.nav-links{
    width: 80vw;
    display: flex;
    padding: 0 0.7vw;
    justify-content: space-evenly;
    align-items: center;
    text-transform: uppercase;
    list-style: none;
    font-weight: 600;
}

.nav-links li a{
    margin: 0 0.7vw;
    text-decoration: none;
    transition: all ease-in-out 350ms;
    padding: 10px;
}

.nav-links li a:hover{
    color:#000;
    background-color: #fff;
    padding: 10px;
    border-radius: 50px;
}
```

```css
.nav-links li{
    position:relative;
}

.nav-links li a:hover::before{
    width: 80%;
}


/*Buttons Styling*/

.login-button{
    padding: 0.6rem 0.8rem;
    margin-left: 2vw;
    font-size:1rem;
    cursor:pointer;
    background-color: transparent;
    border:1.5px solid #f2f5f7;
    border-radius: 2em;
}

.login-button:hover{
    color:#fff;
    background-color: #dd5f24;
    border:1.5px solid #dd5f24;
    transition: all ease-in-out 350ms;
}

/*Navigation Icon Styling*/

.hamburger div{
    width: 30px;
    height: 3px;
    background: #f2f5f7;
    margin: 5px;
    transition: all 0.3s ease;
}

.hamburger{
    display: none;
}


/*Responsive Adding Media Queries*/

@media screen and (max-width: 800px){
    nav{
        position: fixed;
        z-index: 3;
```

```
}
.hamburger{
    display:block;
    position: absolute;
    cursor: pointer;
    right: 5%;
    top: 50%;
    transform: translate(-5%, -50%);
    z-index: 2;
    transition: all 0.7s ease;
}
.nav-links{
    background: #053742;
    position: fixed;
    opacity: 1;
    height: 100vh;
    width: 100%;
    flex-direction: column;
    clip-path: circle(50px at 90% -20%);
    -webkit-clip-path: circle(50px at 90% -10%);
    transition: all 1s ease-out;
    pointer-events: none;
}
.nav-links.open{
    clip-path: circle(1000px at 90% -10%);
    -webkit-clip-path: circle(1000px at 90% -10%);
    pointer-events: all;
}
.nav-links li{
    opacity: 0;
}
.nav-links li:nth-child(1){
    transition: all 0.5s ease 0.2s;
}
.nav-links li:nth-child(2){
    transition: all 0.5s ease 0.4s;
}
.nav-links li:nth-child(3){
    transition: all 0.5s ease 0.6s;
}
.nav-links li:nth-child(4){
    transition: all 0.5s ease 0.7s;
}
.nav-links li:nth-child(5){
    transition: all 0.5s ease 0.8s;
}
.nav-links li:nth-child(6){
    transition: all 0.5s ease 0.9s;
```

```
        margin: 0;
    }
    .nav-links li:nth-child(7){
        transition: all 0.5s ease 1s;
        margin: 0;
    }

    li.fade{
        opacity: 1;
    }

    /* Navigation Bar Icon on Click*/

        .toggle .bars1{
            transform: rotate(-45deg) translate(-5px, 6px);
        }

        .toggle .bars2{
            transition: all 0s ease;
            width: 0;
        }

        .toggle .bars3{
            transform: rotate(45deg) translate(-5px, -6px);
        }

}
```

JS code :-

```
const hamburger = document.querySelector(".hamburger");
const navLinks = document.querySelector(".nav-links");
const links = document.querySelectorAll(".nav-links li");

hamburger.addEventListener('click', ()=>{
   //Links
    navLinks.classList.toggle("open");
    links.forEach(link => {
        link.classList.toggle("fade");
    });

    //Animation
    hamburger.classList.toggle("toggle");
});
```

4). Create a form and validate using JavaScript :-

HTML code :-

```
<!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>Document</title>
    <link rel="stylesheet" href="style2.css">
  </head>
  <body>
    <h1 style="text-align: center">REGISTRATION FORM</h1>
    <form name="RegForm" onsubmit="return Registration()" method="post">
      <p>Name: <input type="text" size="65" name="Name" /></p>

      <br />

      <p>Address: <input type="text" size="65" name="Address" /></p>

      <br />

      <p>E-mail Address: <input type="text" size="65" name="EMail" /></p>

      <br />

      <p>Password: <input type="text" size="65" name="Password" /></p>

      <br />

      <p>Telephone: <input type="text" size="65" name="Telephone" /></p>

      <br />

      <p>
        SELECT YOUR COURSE
        <select type="text" value="" name="Subject">
          <option>BTECH</option>
          <option>BBA</option>
          <option>BCA</option>
          <option>B.COM</option>
        </select>
      </p>

      <br />
      <br />

      <p>Comments: <textarea cols="55" name="Comment"> </textarea></p>
```

```
      <p>
        <input type="submit" value="send" name="Submit" />
        <input type="reset" value="Reset" name="Reset" />
      </p>
    </form>
    <script src="script2.js"></script>
  </body>
</html>
```

CSS code :-

```
div
{
    box-sizing: border-box;
    width: 100%;
    border: 100px solid black;
    float: left;
    align-content: center;
    align-items: center;
}

form
{
    margin: 0 auto;
    width: 600px;
}
```

JS code :-

```
            function Registration() {
                var name =
                    document.forms.RegForm.Name.value;
                var email =
                    document.forms.RegForm.EMail.value;
                var phone =
                    document.forms.RegForm.Telephone.value;
                var what =
                    document.forms.RegForm.Subject.value;
                var password =
                    document.forms.RegForm.Password.value;
                var address =
                    document.forms.RegForm.Address.value;
```

```
                var regEmail=/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/g;
//Javascript reGex for Email Validation.
                var
regPhone=/^\d{10}$/;                                    // Javascript reGex
for Phone Number validation.
                var regName = /\d+$/g;                                  //
Javascript reGex for Name validation

                if (name == "" || regName.test(name)) {
                    window.alert("Please enter your name properly.");
                    name.focus();
                    return false;
                }

                if (address == "") {
                    window.alert("Please enter your address.");
                    address.focus();
                    return false;
                }

                if (email == "" || !regEmail.test(email)) {
                    window.alert("Please enter a valid e-mail address.");
                    email.focus();
                    return false;
                }

                if (password == "") {
                    alert("Please enter your password");
                    password.focus();
                    return false;
                }

                if(password.length <6){
                    alert("Password should be atleast 6 character long");
                    password.focus();
                    return false;

                }
                if (phone == "" || !regPhone.test(phone)) {
                    alert("Please enter valid phone number.");
                    phone.focus();
                    return false;
                }

                if (what.selectedIndex == -1) {
                    alert("Please enter your course.");
                    what.focus();
                    return false;
```

```
            }

            return true;
        }
```

5). Create a model box using CSS and JS with three buttons :-

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <style>
      body {
        font-family: Arial, Helvetica, sans-serif;
      }

      /* The Modal (background) */
      .modal {
        display: none; /* Hidden by default */
        position: fixed; /* Stay in place */
        z-index: 1; /* Sit on top */
        padding-top: 100px; /* Location of the box */
        left: 0;
        top: 0;
        width: 100%; /* Full width */
        height: 100%; /* Full height */
        overflow: auto; /* Enable scroll if needed */
        background-color: rgb(0, 0, 0); /* Fallback color */
        background-color: rgba(0, 0, 0, 0.4); /* Black w/ opacity */
      }

      /* Modal Content */
      .modal-content {
        background-color: #fefefe;
        margin: auto;
        padding: 20px;
        border: 1px solid #888;
        width: 80%;
      }

      /* The Close Button */
      .close {
        color: #aaaaaa;
        float: right;
        font-size: 28px;
        font-weight: bold;
      }
```

```
    .close:hover,
    .close:focus {
      color: #000;
      text-decoration: none;
      cursor: pointer;
    }
  </style>
</head>
<body>
  <h2>Modal Example</h2>

  <!-- Trigger/Open The Modal -->
  <button id="myBtn">Open Modal</button>

  <!-- The Modal -->
  <div id="myModal" class="modal">
    <!-- Modal content -->
    <div class="modal-content">
      <span class="close">&times;</span>
      <p>Some text in the Modal..</p>
    </div>
  </div>

  <script>
    // Get the modal
    var modal = document.getElementById("myModal");

    // Get the button that opens the modal
    var btn = document.getElementById("myBtn");

    // Get the <span> element that closes the modal
    var span = document.getElementsByClassName("close")[0];

    // When the user clicks the button, open the modal
    btn.onclick = function () {
      modal.style.display = "block";
    };

    // When the user clicks on <span> (x), close the modal
    span.onclick = function () {
      modal.style.display = "none";
    };

    // When the user clicks anywhere outside of the modal, close it
    window.onclick = function (event) {
      if (event.target == modal) {
        modal.style.display = "none";
      }
```

```
      };
    </script>
  </body>
</html>
```

6). Using external JS library to show slider :-

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>My Slider</title>
    <style type="text/css">
      body {
        margin: 0;
        background: #e6e6e6;
      }
      .showSlide {
        display: none;
      }
      .showSlide img {
        width: 100%;
        height: 500px;
      }
      .slidercontainer {
        max-width: 1000px;
        position: relative;
        margin: auto;
      }
      .left,
      .right {
        cursor: pointer;
        position: absolute;
        top: 50%;
        width: auto;
        padding: 16px;
        margin-top: -22px;
        color: white;
        font-weight: bold;
        font-size: 18px;
        transition: 0.6s ease;
        border-radius: 0 3px 3px 0;
      }
      .right {
        right: 0;
        border-radius: 3px 0 0 3px;
```

```
    }
    .left:hover,
    .right:hover {
      background-color: rgba(115, 115, 115, 0.8);
    }
    .content {
      color: #eff5d4;
      font-size: 30px;
      padding: 8px 12px;
      position: absolute;
      top: 10px;
      width: 100%;
      text-align: center;
    }
    .active {
      background-color: #717171;
    }
    /* Fading animation */
    .fade {
      -webkit-animation-name: fade;
      -webkit-animation-duration: 1.5s;
      animation-name: fade;
      animation-duration: 1.5s;
    }
    @-webkit-keyframes fade {
      from {
        opacity: 0.4;
      }
      to {
        opacity: 1;
      }
    }

    @keyframes fade {
      from {
        opacity: 0.4;
      }
      to {
        opacity: 1;
      }
    }
  </style>
</head>
<body>
  <div class="slidercontainer">
    <div class="showSlide fade">
      <img src="Images/adi-goldstein-1Xafaerat1s-unsplash.jpg" />
      <div class="content">Slide1 heading</div>
```

```
      </div>
      <div class="showSlide fade">
        <img src="Images/adi-goldstein-JTlwuJblZdk-unsplash.jpg" />
        <div class="content">Slide2 heading</div>
      </div>

      <div class="showSlide fade">
        <img src="Images/alexander-milo-HwxVLhLyg2s-unsplash.jpg" />
        <div class="content">Slide3 heading</div>
      </div>
      <div class="showSlide fade">
        <img src="Images/istockphoto-914886270-1024x1024.jpg" />
        <div class="content">Slide4 heading</div>
      </div>
      <!-- Navigation arrows -->
      <a class="left" onclick="nextSlide(-1)">❮</a>
      <a class="right" onclick="nextSlide(1)">❯</a>
    </div>

    <script type="text/javascript">
      var slide_index = 1;
      displaySlides(slide_index);

      function nextSlide(n) {
        displaySlides((slide_index += n));
      }

      function currentSlide(n) {
        displaySlides((slide_index = n));
      }

      function displaySlides(n) {
        var i;
        var slides = document.getElementsByClassName("showSlide");
        if (n > slides.length) {
          slide_index = 1;
        }
        if (n < 1) {
          slide_index = slides.length;
        }
        for (i = 0; i < slides.length; i++) {
          slides[i].style.display = "none";
        }
        slides[slide_index - 1].style.display = "block";
      }
    </script>
  </body>
</html>
```

Module 4 (New Request) :-

1).What is JSON?

JSON stands for JavaScript Object Notation

JSON is a lightweight format for storing and transporting data

JSON is often used when data is sent from a server to a web page

JSON is "self-describing" and easy to understand

2). What is promises?

"Producing code" is code that can take some time

"Consuming code" is code that must wait for the result

A Promise is a JavaScript object that links producing code and consuming code

3). Write a program of promises and handle that promises also

```
var promise = new Promise(function(resolve, reject) {
  const x = "jainikforjainik";
  const y = "jainikforjainik"
  if(x === y) {
    resolve();
  } else {
    reject();
  }
});

promise.
    then(function () {
        console.log('Success, You are a JAINIK);
```

```
}).
catch(function () {
    console.log('Some error has occurred');
});
```

4). Use fetch method for calling an api https://fakestoreapi.com/products

fakeStoreApi can be used with any type of shopping project that needs products, carts, and users in JSON format. you can use examples below to check how fakeStoreApi works

```
fetch('https://fakestoreapi.com/products')
        .then(res=>res.json())
        .then(json=>console.log(json))
```

output:

```
[
{
Id:1,
Title:'….',
Price:'….',
Category:'….',
Description:'….',
Image:'….'
}
/*…..*/
{
Id:30,
Title:'….',
Price:'….',
Category:'….',
Description:'….',
```

Image:’….’

}

]

5). Display all the product from the api in your HTML page

How to display api**: .**

```
<script>
    function fetchdata() {
    $.get("http://10.10.35.138:5000/data", function (data) { //The link of this line is my api link
            $("#visitor").html('Visitor Count : ' + data.people);
            $("#time").html('Time : ' + data.time);
        });
    }
</script>
```

****HTML PART****

```
<div class="details">
<p id="visitor">Person Count:</p>
<p id="time">Time:</p>`enter code here`
</div>
```

Display API Data in Html:

```
<!DOCTYPE html>
<html>
<body>
<h1>API Data</h1>
<div id="container">
   <div id="api">Nothing Yet</div>
```

```
        </div>

        <br>

        <button type="button" onclick="loadAPI()">Change Content</button>

        <script>
        function loadAPI() {
          var xhttp = new XMLHttpRequest();
          xhttp.open("GET", "API URL with Token here", false);
          xhttp.addEventListener("load", loadData);
          xhttp.send();
        }

        function loadData() {
          document.getElementById('api').innerText = JSON.parse(this.responseText);
        }
        </script>

        </body>
</html>
```