Assignment 4.2

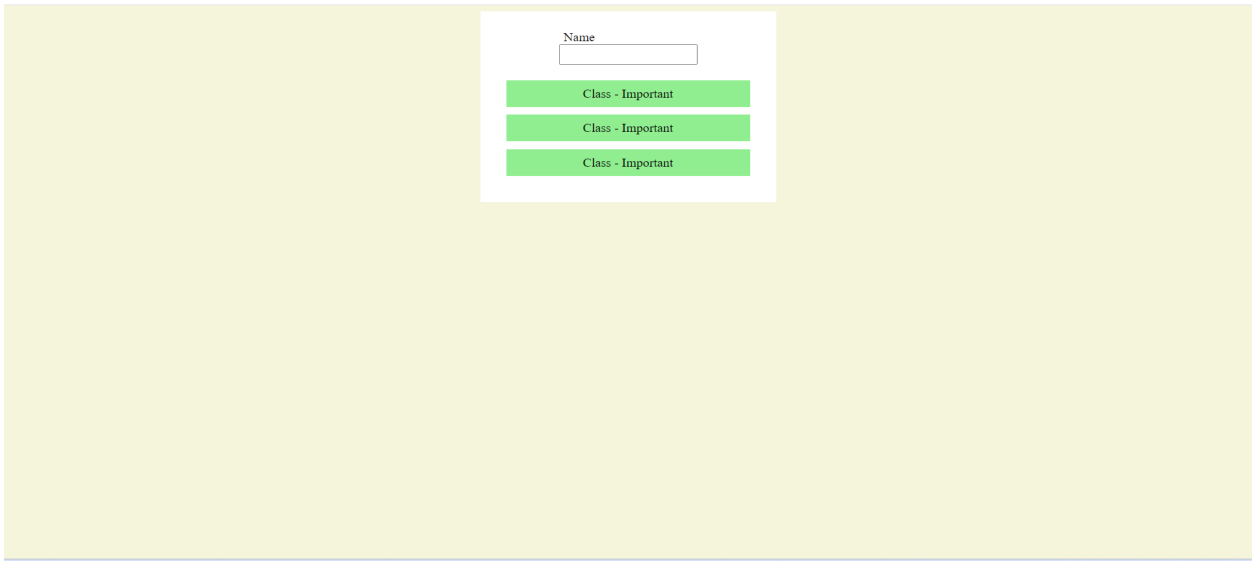
Write and submit code to set up a handler for a focus - event on an input element having id name which is a direct child of an element having id john.

- When run the handler should add the HTML class shape to all elements having a HTML class of important which are direct children of an element having id john.
- The event should not propagate beyond the handler.

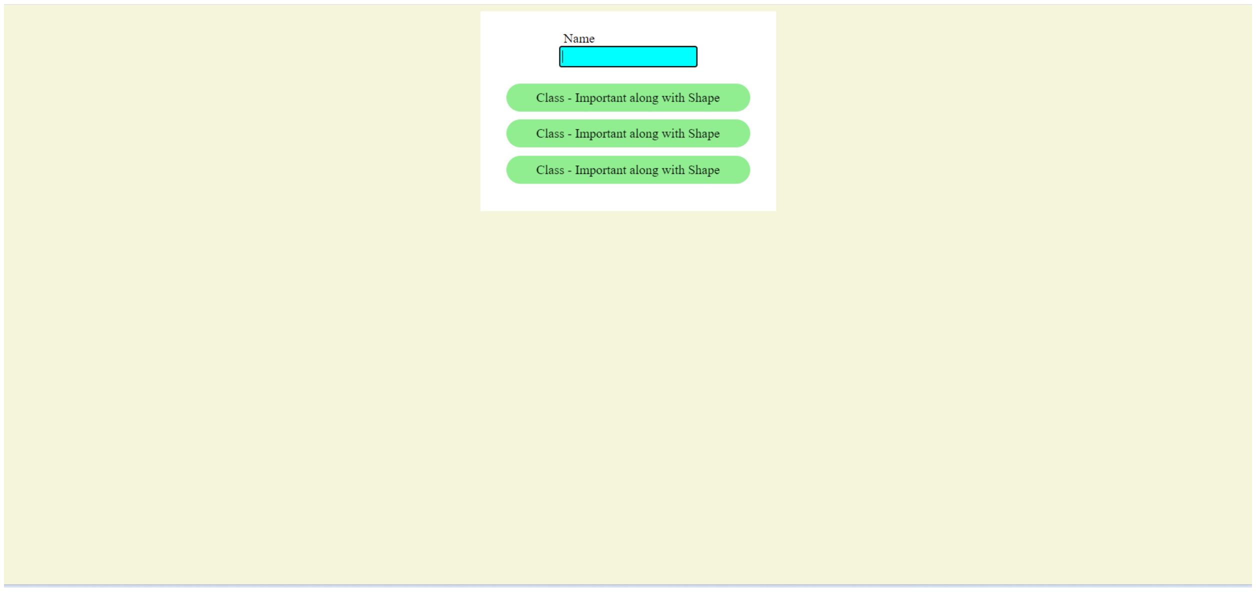
Code:

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title>Bryan Lewis - Assessment SDE</title>

  <style>
    body {
      background-color: beige;
    }

    .important {
      margin: 0 auto;
      margin-bottom: 10px;
      text-align:center;
      background-color:lightgreen;
      padding: 0.5rem 0;
    }

    #john {
      width: 300px;
      padding: 1.5rem 2.0rem;
      margin: 0 auto;
      margin-left: auto;
      margin-right: auto;
      background-color: white;
    }
    .shape {
      border-radius: 30px;
    }
    .inputdiv {
            text-align: center;
        }
        .inputdiv > input {
            text-align: left;
        }
        input {
            height: 20px;
            margin-bottom: 20px;
        }
        label {
            text-align:center;
            margin-right: 120px;
        }

    </style>
</head>
<body>
    <div id="john">
        <label style="display:block;" for="name">Name</label>
        <div class="inputdiv">
            <input id="name" type="text" />
        </div>
        <div class="important">Class - Important</div>
        <div class="important">Class - Important</div>
        <div class="important">Class - Important</div>
    </div>

    <script type="text/javascript">

        const name = document.getElementById('name');

        name.addEventListener('focus', (event) => {
            event.stopPropagation();
            event.target.style.background = 'aqua';
        });

        name.addEventListener('focus', (event) => {
            let child = document.getElementsByClassName('important');
            for (let elements of child) {
                elements.classList.add('shape');
            elements.innerText = "Class - Important along with Shape";
          }
        });

        name.addEventListener('blur', (event) => {
          event.stopPropagation();
          event.target.style.background = '';
        });

        name.addEventListener('blur', (event) => {
          let child = document.getElementsByClassName('important');
          for (let elements of child) {
            elements.classList.remove('shape');
            elements.innerText = "Class - Important";
          }
        });
    </script>
</body>
</html>
```

Output - Blur Event



Output - Focus Event


Name
Class - Important along with Shape
Class - Important along with Shape
Class - Important along with Shape

1.4 Explain JWT in approx and prove its top 3 benefits

JSON Web Token (JWT) is an open standard that defines the compact and self - contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted as it is digitally signed. The tokens are signed either using a private secret or a public/private key.

JWT relies on other JSON - based standards: JSON Web Signature and JSON Web Encryption.

The benefits of JWT are -

- More compact - JSON is less verbose than XML, so when it is encoded, a JWT is smaller than a Security Assertion Markup Language (SAML). This makes JWT a good choice to be passed in HTML and HTTP environments.
- More secure - JWTs can use a public/private key in the form of an X.509 certificate for signing. A JWT can also be symmetrically signed by a shared secret using the HMAC algorithm.
- More common - JSON parsers are common in most programming languages because they map directly to objects. Conversely, XML doesn't have a natural document - to - object mapping. This makes it easier to work with JWT than SAML assertions.

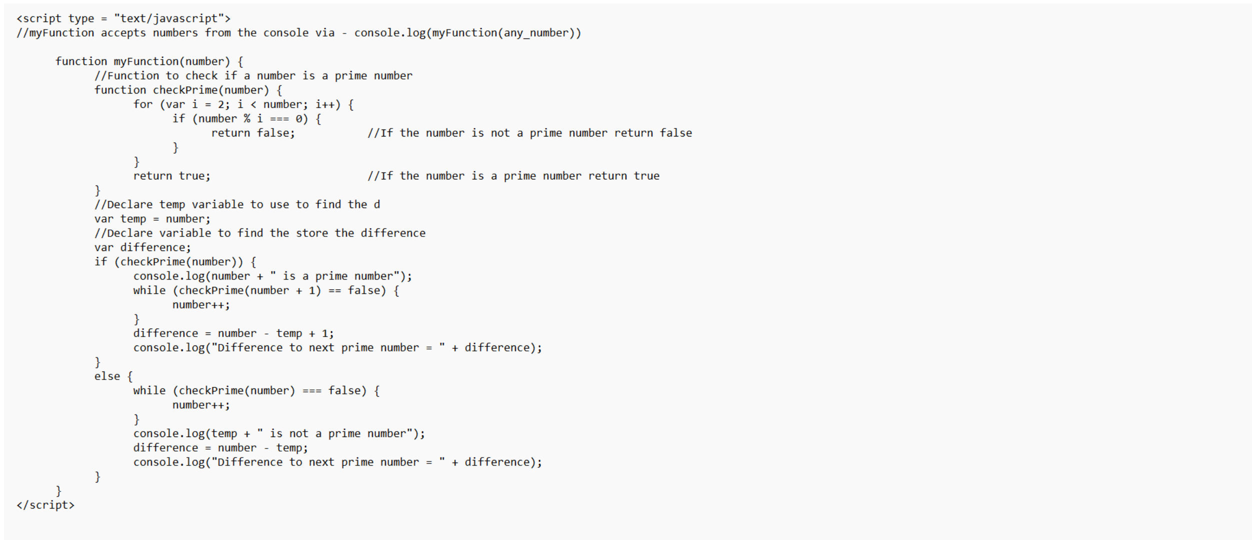
8.2 Given a number x, find out if it is a prime number or not, use javascript and find out the difference between the next prime number after x and x.

Code:

```
<script type = "text/javascript">
//myFunction accepts numbers from the console via - console.log(myFunction(any_number))

        function myFunction(number) {
                //Function to check if a number is a prime number
                function checkPrime(number) {
                        for (var i = 2; i < number; i++) {
                                if (number % i === 0) {
                                        return false;           //If the number is not a prime
number return false
                                }
                        }
                        return true;                            //If the number is a prime number
return true
                }
                //Declare temp variable to use to find the d
                var temp = number;
                //Declare variable to find the store the difference
                var difference;
                if (checkPrime(number)) {
                        console.log(number + " is a prime number");
                        while (checkPrime(number + 1) == false) {
                                number++;
                        }
                        difference = number - temp + 1;
                        console.log("Difference to next prime number = " + difference);
                }
                else {
                        while (checkPrime(number) === false) {
                        number++;
                        }
                        console.log(temp + " is not a prime number");
                        difference = number - temp;
                        console.log("Difference to next prime number = " + difference);
                }
        }
```

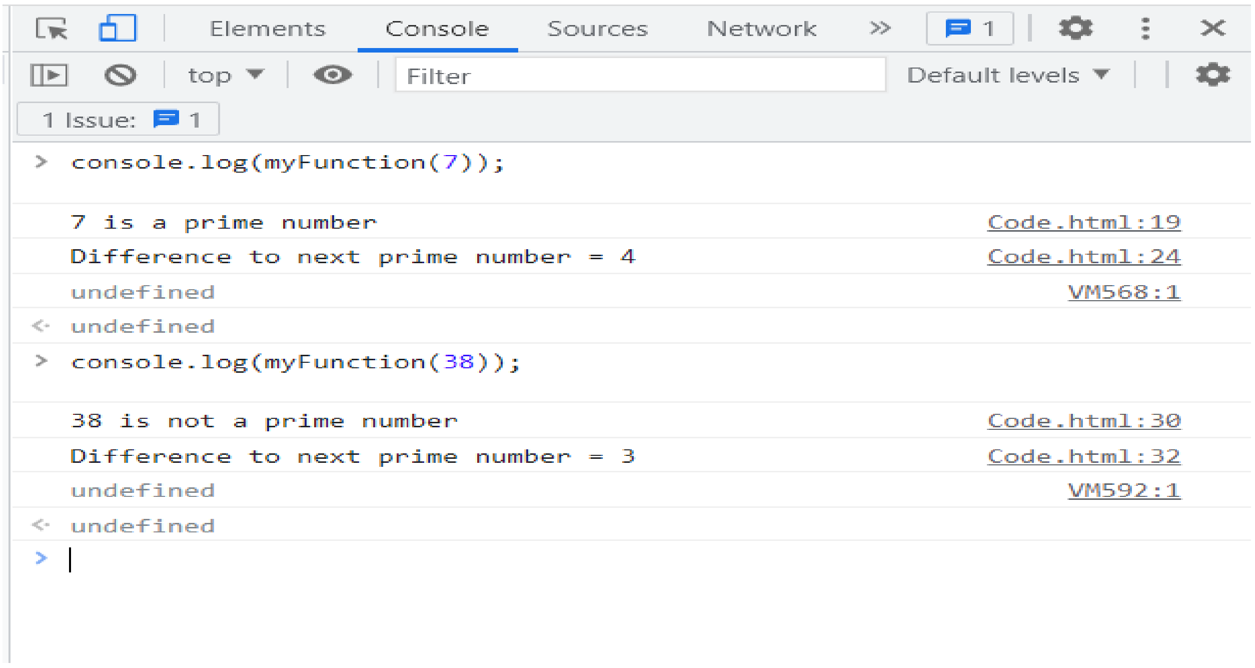
</script>

Code Screenshot:



```
<script type = "text/javascript">
//myFunction accepts numbers from the console via - console.log(myFunction(any_number))

      function myFunction(number) {
            //Function to check if a number is a prime number
            function checkPrime(number) {
                  for (var i = 2; i < number; i++) {
                        if (number % i === 0) {
                              return false;           //If the number is not a prime number return false
                        }
                  }
                  return true;                        //If the number is a prime number return true
            }
            //Declare temp variable to use to find the d
            var temp = number;
            //Declare variable to find the store the difference
            var difference;
            if (checkPrime(number)) {
                  console.log(number + " is a prime number");
                  while (checkPrime(number + 1) == false) {
                        number++;
                  }
                  difference = number - temp + 1;
                  console.log("Difference to next prime number = " + difference);
            }
            else {
                  while (checkPrime(number) === false) {
                        number++;
                  }
                  console.log(temp + " is not a prime number");
                  difference = number - temp;
                  console.log("Difference to next prime number = " + difference);
            }
      }
</script>
```

Code Output:



```
> console.log(myFunction(7));

  7 is a prime number                              Code.html:19
  Difference to next prime number = 4              Code.html:24
  undefined                                          VM568:1
<· undefined
> console.log(myFunction(38));

  38 is not a prime number                         Code.html:30
  Difference to next prime number = 3              Code.html:32
  undefined                                          VM592:1
<· undefined
>
```