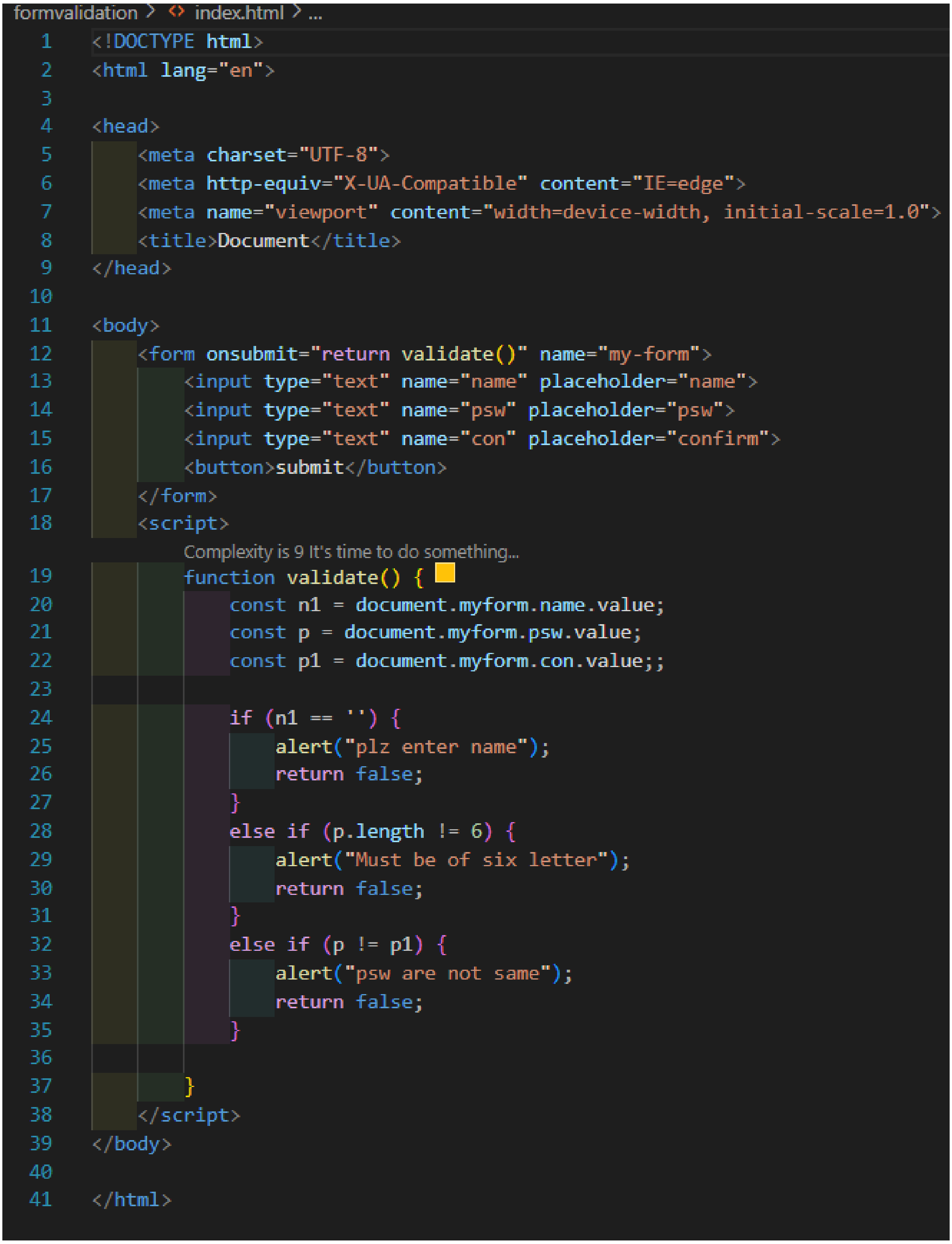
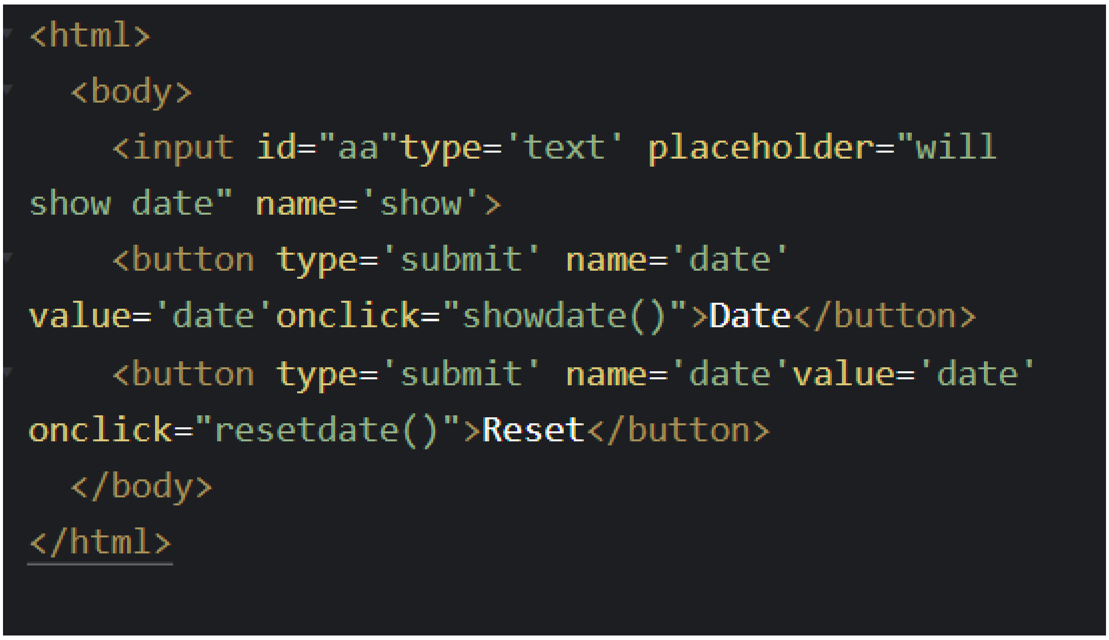
Q.1 (A)

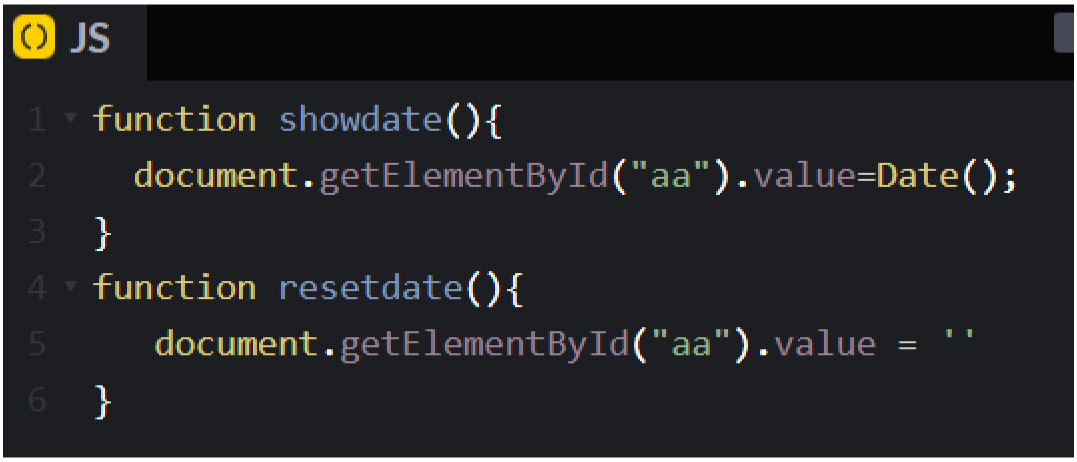

formvalidation > <> index.html > ...
Complexity is 9 It's time to do something...

```
<!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>
    <form onsubmit="return validate()" name="my-form">
        <input type="text" name="name" placeholder="name">
        <input type="text" name="psw" placeholder="psw">
        <input type="text" name="con" placeholder="confirm">
        <button>submit</button>
    </form>
    <script>
        function validate() {
            const n1 = document.myform.name.value;
            const p = document.myform.psw.value;
            const p1 = document.myform.con.value;;

            if (n1 == '') {
                alert("plz enter name");
                return false;
            }
            else if (p.length != 6) {
                alert("Must be of six letter");
                return false;
            }
            else if (p != p1) {
                alert("psw are not same");
                return false;
            }

        }
    </script>
</body>

</html>
```

Q.3 (A)

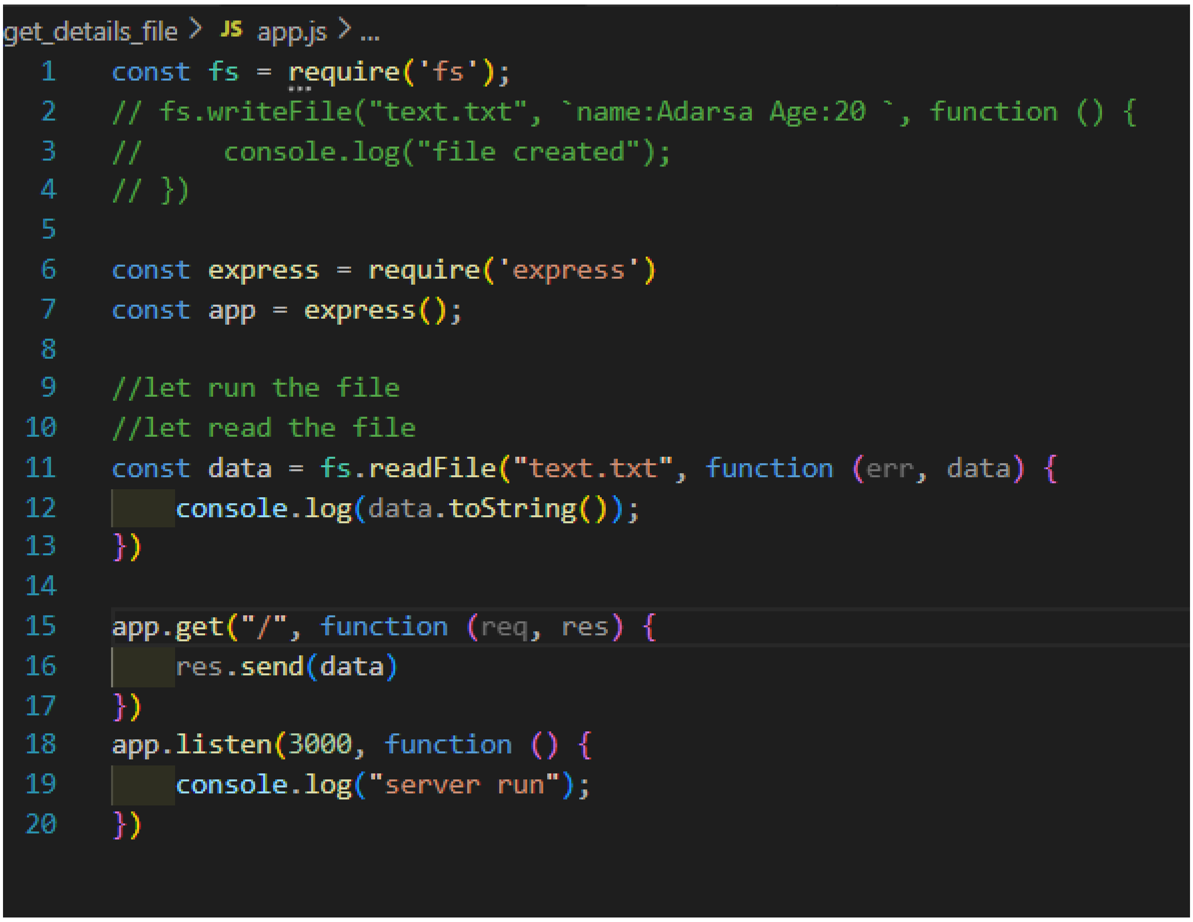


```html
<html>
  <body>
    <input id="aa"type='text' placeholder="will
show date" name='show'>
    <button type='submit' name='date'
value='date'onclick="showdate()">Date</button>
    <button type='submit' name='date'value='date'
onclick="resetdate()">Reset</button>
  </body>
</html>
```



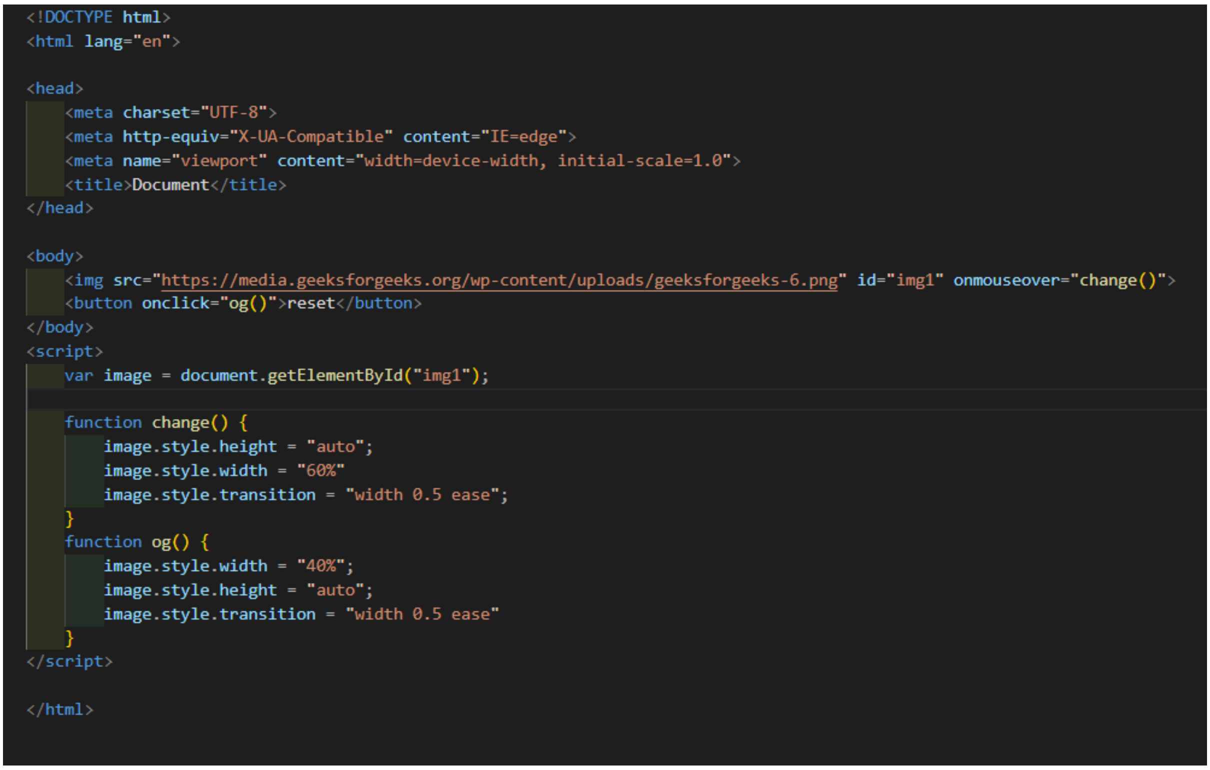
```javascript
function showdate(){
  document.getElementById("aa").value=Date();
}
function resetdate(){
   document.getElementById("aa").value = ''
}
```

Q3 (B)



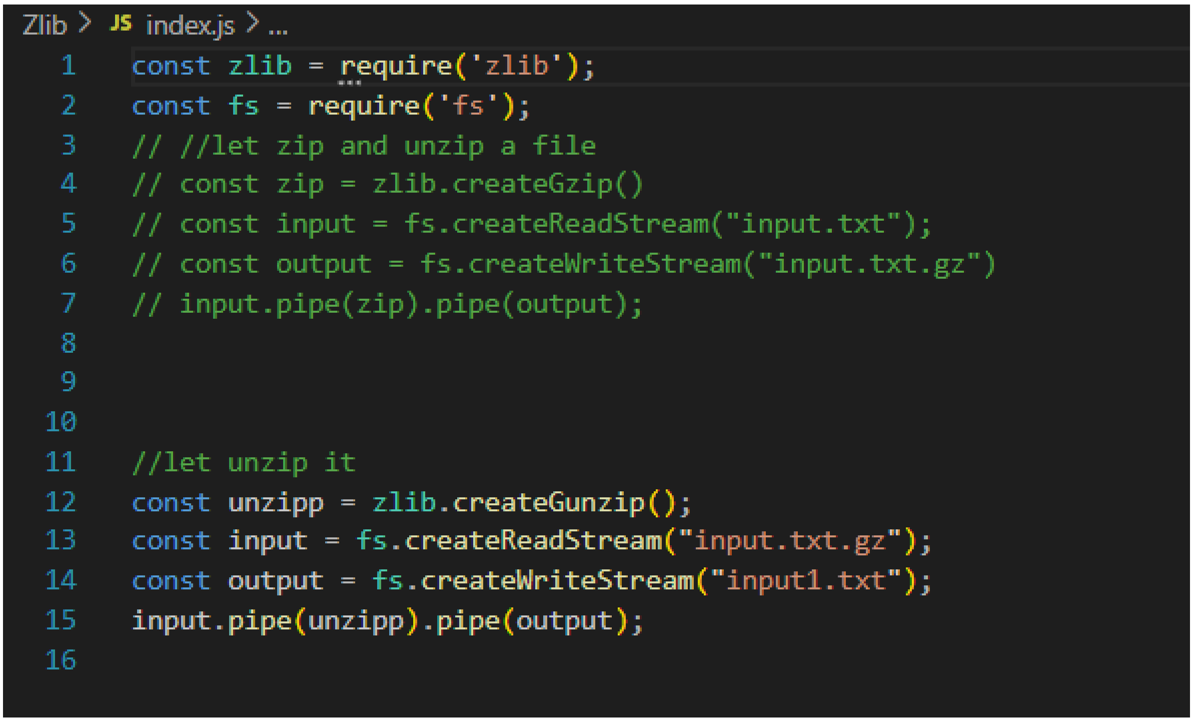
```
const fs = require('fs');
// fs.writeFile("text.txt", `name:Adarsa Age:20 `, function () {
//      console.log("file created");
// })

const express = require('express')
const app = express();

//let run the file
//let read the file
const data = fs.readFile("text.txt", function (err, data) {
    console.log(data.toString());
})

app.get("/", function (req, res) {
    res.send(data)
})
app.listen(3000, function () {
    console.log("server run");
})
```

Q.4 (A)



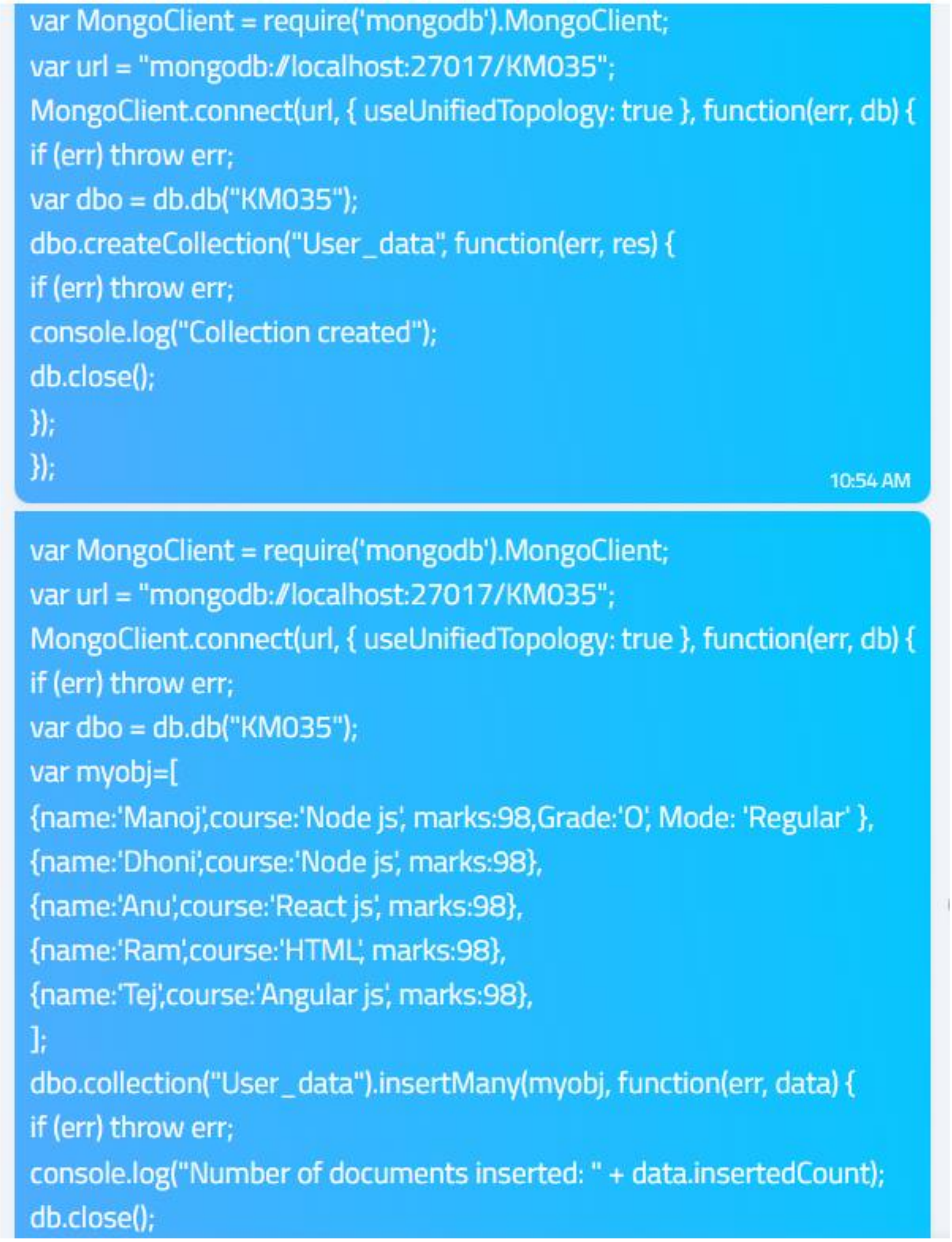
```
<!DOCTYPE html>
<html lang="en">

<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>
    <img src="https://media.geeksforgeeks.org/wp-content/uploads/geeksforgeeks-6.png" id="img1" onmouseover="change()">
    <button onclick="og()">reset</button>
</body>
<script>
    var image = document.getElementById("img1");

    function change() {
        image.style.height = "auto";
        image.style.width = "60%"
        image.style.transition = "width 0.5 ease";
    }
    function og() {
        image.style.width = "40%";
        image.style.height = "auto";
        image.style.transition = "width 0.5 ease"
    }
</script>

</html>
```

Q.4 (B)



```
const zlib = require('zlib');
const fs = require('fs');
// //let zip and unzip a file
// const zip = zlib.createGzip()
// const input = fs.createReadStream("input.txt");
// const output = fs.createWriteStream("input.txt.gz")
// input.pipe(zip).pipe(output);



//let unzip it
const unzipp = zlib.createGunzip();
const input = fs.createReadStream("input.txt.gz");
const output = fs.createWriteStream("input1.txt");
input.pipe(unzipp).pipe(output);

```

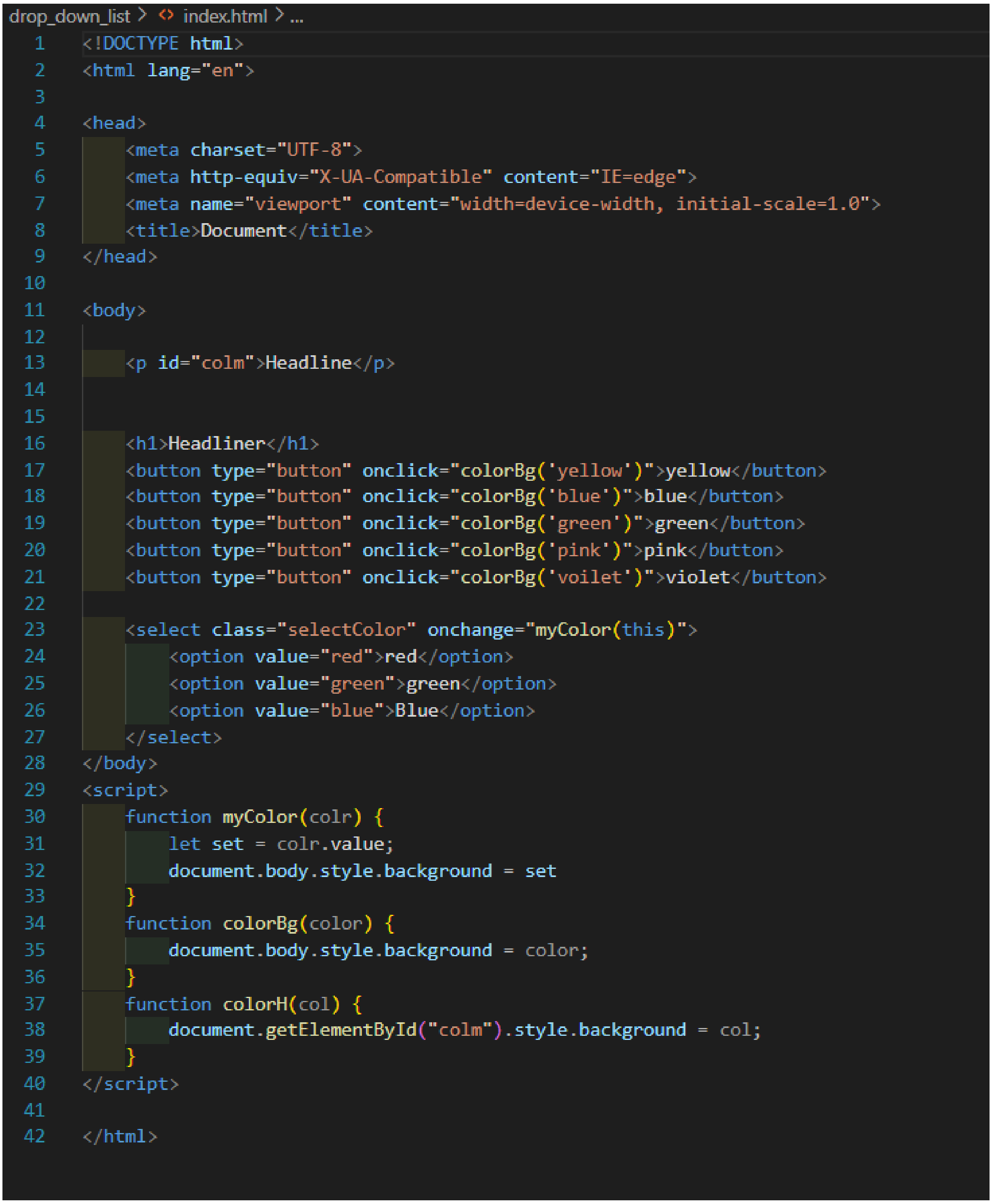
Q.5(b)



```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/KM035";
MongoClient.connect(url, { useUnifiedTopology: true }, function(err, db) {
if (err) throw err;
var dbo = db.db("KM035");
dbo.createCollection("User_data", function(err, res) {
if (err) throw err;
console.log("Collection created");
db.close();
});
});
```

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/KM035";
MongoClient.connect(url, { useUnifiedTopology: true }, function(err, db) {
if (err) throw err;
var dbo = db.db("KM035");
var myobj=[
{name:'Manoj',course:'Node js', marks:98,Grade:'O', Mode: 'Regular' },
{name:'Dhoni',course:'Node js', marks:98},
{name:'Anu',course:'React js', marks:98},
{name:'Ram',course:'HTML', marks:98},
{name:'Tej',course:'Angular js', marks:98},
];
dbo.collection("User_data").insertMany(myobj, function(err, data) {
if (err) throw err;
console.log("Number of documents inserted: " + data.insertedCount);
db.close();
```

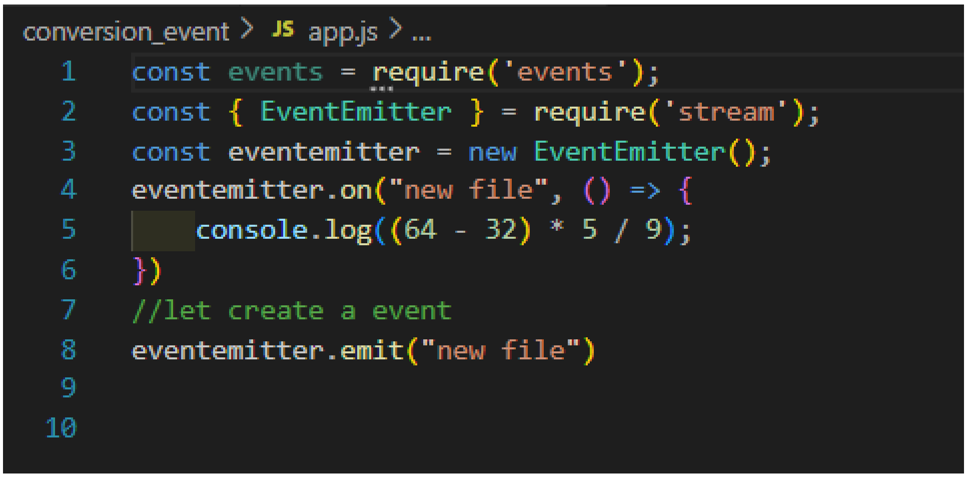
Q.6(A)



drop_down_list > index.html > ...

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

<body>

    <p id="colm">Headline</p>


    <h1>Headliner</h1>
    <button type="button" onclick="colorBg('yellow')">yellow</button>
    <button type="button" onclick="colorBg('blue')">blue</button>
    <button type="button" onclick="colorBg('green')">green</button>
    <button type="button" onclick="colorBg('pink')">pink</button>
    <button type="button" onclick="colorBg('voilet')">violet</button>

    <select class="selectColor" onchange="myColor(this)">
        <option value="red">red</option>
        <option value="green">green</option>
        <option value="blue">Blue</option>
    </select>
</body>
<script>
    function myColor(colr) {
        let set = colr.value;
        document.body.style.background = set
    }
    function colorBg(color) {
        document.body.style.background = color;
    }
    function colorH(col) {
        document.getElementById("colm").style.background = col;
    }
</script>

</html>
```

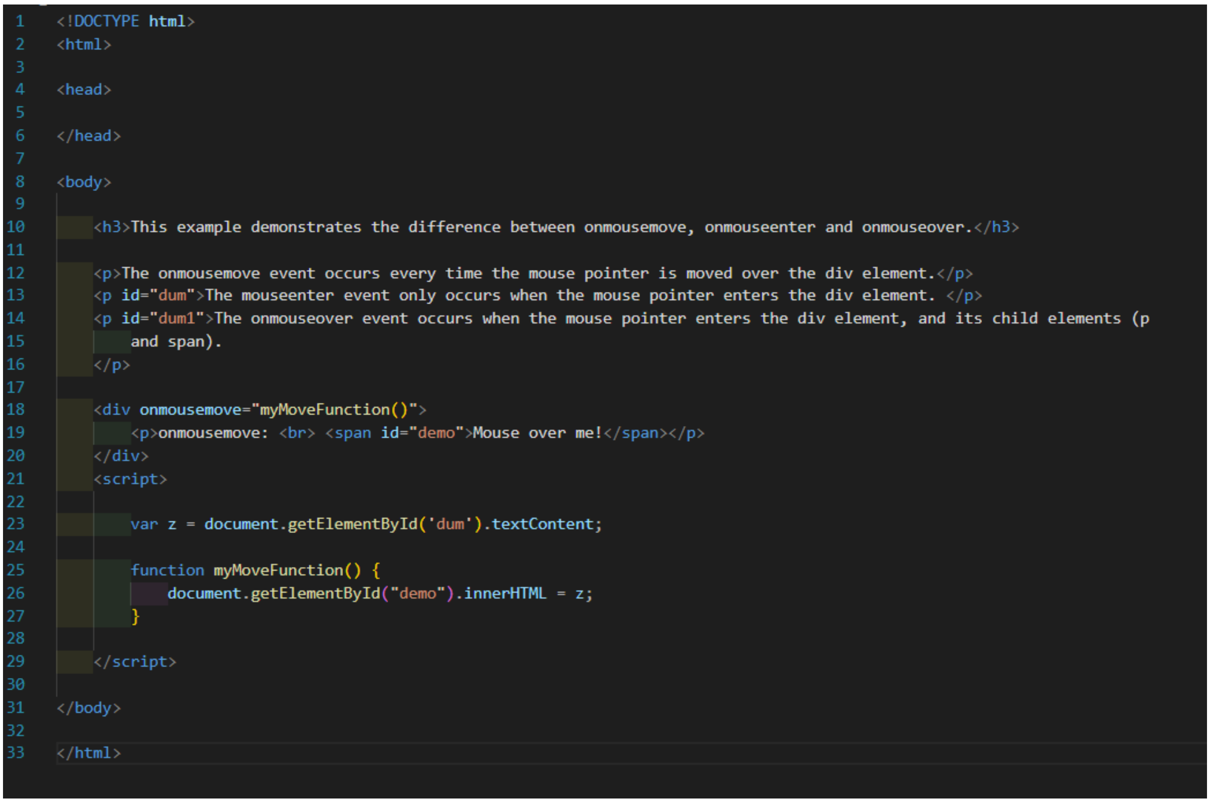
<u>Q.6(B)</u>



conversion_event > JS app.js > ...

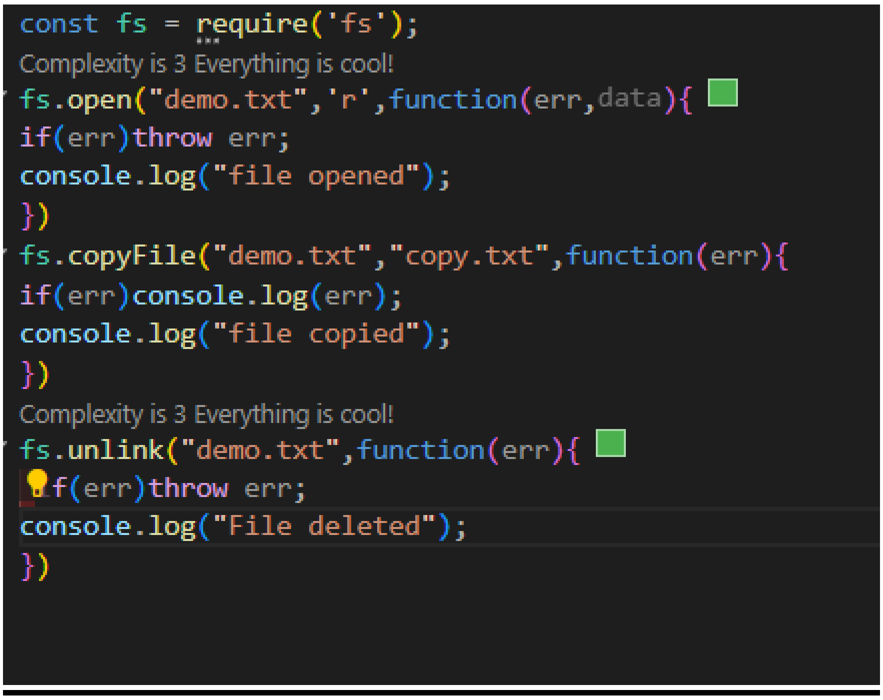
```javascript
const events = require('events');
const { EventEmitter } = require('stream');
const eventemitter = new EventEmitter();
eventemitter.on("new file", () => {
    console.log((64 - 32) * 5 / 9);
})
//let create a event
eventemitter.emit("new file")


```

Q.8(A)



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

<head>

</head>

<body>

    <h3>This example demonstrates the difference between onmousemove, onmouseenter and onmouseover.</h3>

    <p>The onmousemove event occurs every time the mouse pointer is moved over the div element.</p>
    <p id="dum">The mouseenter event only occurs when the mouse pointer enters the div element. </p>
    <p id="dum1">The onmouseover event occurs when the mouse pointer enters the div element, and its child elements (p
        and span).
    </p>

    <div onmousemove="myMoveFunction()">
        <p>onmousemove: <br> <span id="demo">Mouse over me!</span></p>
    </div>
    <script>

        var z = document.getElementById('dum').textContent;

        function myMoveFunction() {
            document.getElementById("demo").innerHTML = z;
        }

    </script>

</body>

</html>
```

Q.9(B)



```
const fs = require('fs');
Complexity is 3 Everything is cool!
fs.open("demo.txt",'r',function(err,data){ 🟩
if(err)throw err;
console.log("file opened");
})
fs.copyFile("demo.txt","copy.txt",function(err){
if(err)console.log(err);
console.log("file copied");
})
Complexity is 3 Everything is cool!
fs.unlink("demo.txt",function(err){ 🟩
💡f(err)throw err;
console.log("File deleted");
})
```