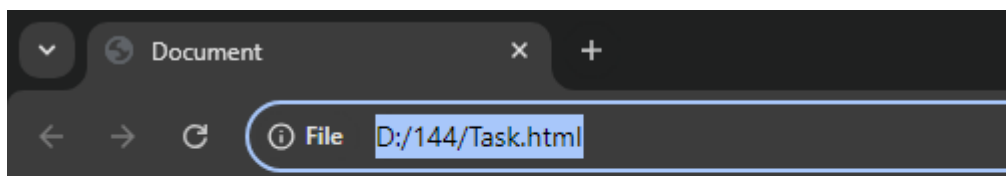


### 3. Closure

#### Task 1:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    function MNC()
    {
      let Manager = 57;
      let Employee = 579;
      function Count()
      {
        document.writeln("Manager: "+Manager+"<br>");
        document.writeln("Employee: "+Employee);
      }
      return Count();
    }
    let TotalNumOfEmployees = MNC();
  </script>
</body>
</html>
```



Manager: 57  
Employee: 579

#### Task 2:

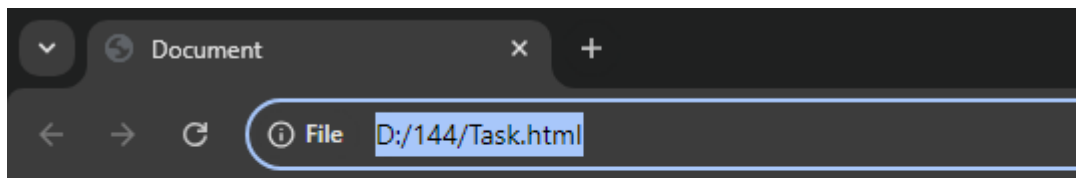
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Counter</title>
</head>
<body>
```

```

<script>
  function Counter()
  {
    let count = 0;

    return function Increament()
    {
      count++;
      document.write("count: "+count+"<br>")
    }
  }
  let Totalcount=Counter();
  Totalcount();
  Totalcount();
  Totalcount();
</Script>
</body>
</html>

```



```

count: 1
count: 2
count: 3

```

### Task 3:

```

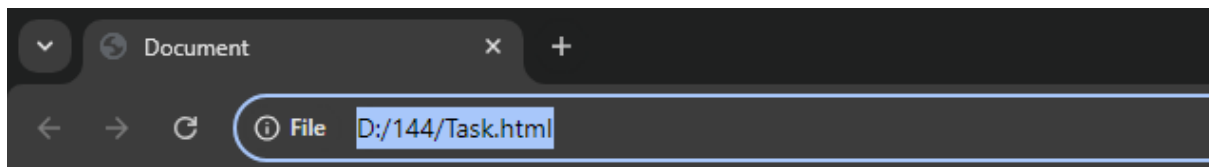
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    function Counter(n)
    {
      let count=n;
      return {
        Increment : function()
        {
          count++;
          document.write("Count: "+count+"<br>");
        },
        Decrement : function()

```

```

    {
        count--;
        document.write("Count: "+count+"<br>");
    }
};
}
let t1count=Counter(3);
let t2count=Counter(1);
let t3count=Counter(9);
t1count.Increment();
t1count.Decrement();
t2count.Increment();
t2count.Decrement();
t3count.Increment();
t3count.Decrement();
</script>
</body>
</html>

```



```

Count: 4
Count: 3
Count: 2
Count: 1
Count: 10
Count: 9

```

#### Task 4:

```

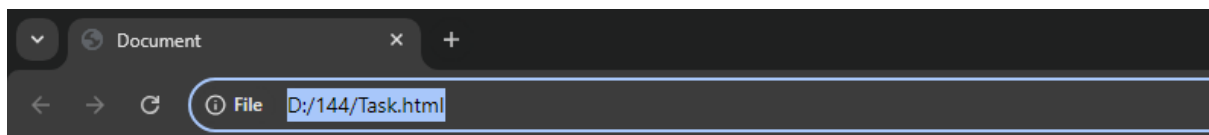
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <script>
        function Counter(n)
        {
            let count=n;
            return {
                Increment : function()
                {
                    count++;
                    document.write("Count: "+count+"<br>");
                }
            }
        }
    </script>

```

```

    },
    Decrement : function()
    {
        count--;
        document.write("Count: "+count+"<br>");
    },
    private : function()
    {
        count = 0;
        document.write("Count: "+count+"<br>");
    }
}
}
let t1count=Counter(3);
let t2count=Counter(1);
let t3count=Counter(9);
t1count.Increment();
t1count.Decrement();
t2count.Increment();
t2count.Decrement();
t3count.Increment();
t3count.Decrement();
t1count.private();
</script>
</body>
</html>

```



Count: 4  
 Count: 3  
 Count: 2  
 Count: 1  
 Count: 10  
 Count: 9  
 Count: 0

### Task 5:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    function Billfactory(Amount)

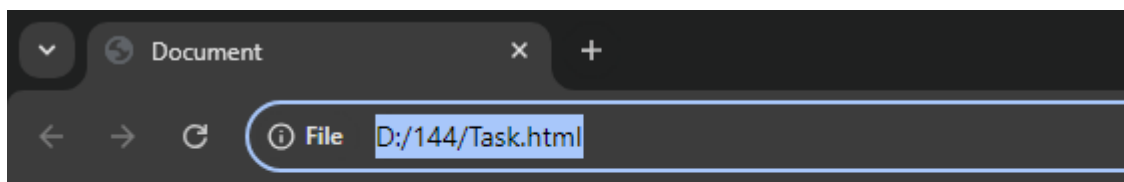
```

```

    {
    if( Amount <= 0)
    {
        return function () {
            document.writeln("Enter the Bill amount above Rs: 0/-"+"<br>");
        }
    }
    else
    {
        return function BillAmount()
        {
            document.writeln("Bill Amount is " +Amount);
        }
    }
    }

    let Result1 = Billfactory(-2);
    Result1();
    let Result2 = Billfactory(5290);
    Result2();
</script>
</body>
</html>

```



Enter the Bill amount above Rs: 0/-  
 Bill Amount is 5290

#### 4. Promise, Promises chaining:

##### Task 1:

```

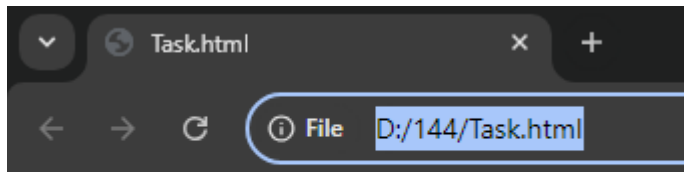
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    let promise = new Promise(function(resolve,reject){
      setTimeout(()=>resolve("Happy Christmas !!!"),5000);
    })

```

```

    promise.then(
      result=>document.writeln(result),
      error=>document.writeln(error)
    );
  </script>
</body>
</html>

```



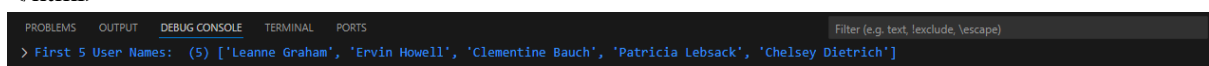
Happy Christmas !!!

## Task 2:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    fetch('https://jsonplaceholder.typicode.com/users')
    .then(response => response.json())
    .then(data => {
      const Fnames = data.slice(0, 5).map(user => user.name);
      return Fnames;
    })
    .then(Fnames => {
      console.log('First 5 User Names: ',Fnames);
    })
    .catch(error => {
      console.log('Error:', error);
    });
  </script>
</body>
</html>

```



## Task 3:

```

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

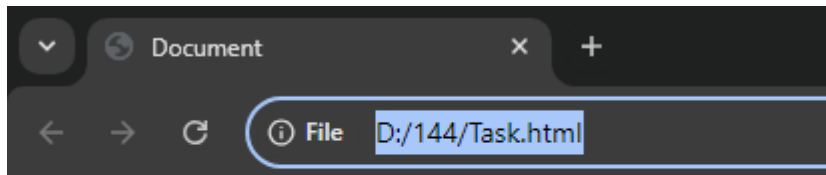
```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
  <script>
    let promise = new Promise(function(resolve,reject){
      const salary = Math.random();
      document.writeln("The random number is: "+salary+"<br>");

      if(salary > 0.1){
        resolve("The number is above 0.1");
      }
      else{
        reject("The number is below 0.1");
      }
    });
    promise.then(
      result=>document.write(result),
      error=>document.write(error)
    );
  </script>
</body>
</html>

```



The random number is: 0.4952001783955564  
 The number is above 0.1

#### Task 4:

```

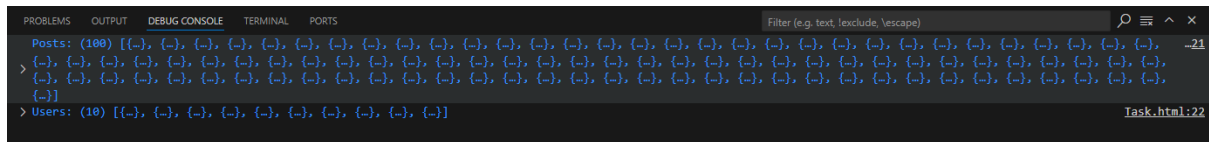
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    function Posts() {
      return fetch('https://jsonplaceholder.typicode.com/posts')
        .then(response => response.json());
    }
    function Users() {

```

```

        return fetch('https://jsonplaceholder.typicode.com/users')
        .then(response => response.json());
    }
    Promise.all([Posts(), Users()])
    .then(results => {
        const [posts, users] = results;
        console.log('Posts:', posts);
        console.log('Users:', users);
    })
    .catch(error => {
        console.log('Error:', error);
    });
</script>
</body>
</html>

```



## Task 5:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <script>
    function UserName() {
      return new Promise((resolve) => {
        setTimeout(() => {
          console.log('User name fetched');
          resolve('Williams');
        }, 1000);
      });
    }
    function UserAge() {
      return new Promise((resolve) => {
        setTimeout(() => {
          console.log('User age fetched');
          resolve(42);
        }, 1000);
      });
    }
    function UserCity() {

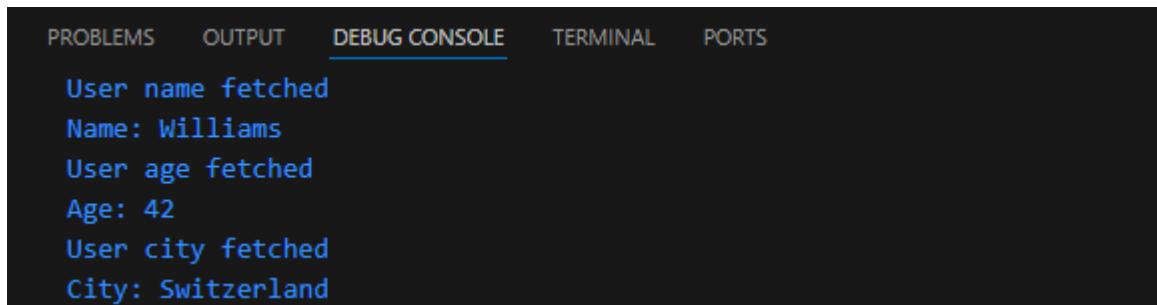
```



```

        return new Promise((resolve) => {
            setTimeout(() => {
                console.log('User city fetched');
                resolve('Switzerland');
            }, 1000);
        });
    }
    UserName()
    .then((name) => {
        console.log('Name:', name);
        return UserAge();
    })
    UserAge()
    .then((age) => {
        console.log('Age:', age);
        return UserCity();
    })
    UserCity()
    .then((city) => {
        console.log('City:', city);
    })
    .catch((error) => {
        console.log('Error:', error);
    });
</script>
</body>
</html>

```



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

User name fetched
Name: Williams
User age fetched
Age: 42
User city fetched
City: Switzerland

```

## 5. Async/await:

### Task 1:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

```

```
<script>
function UserName() {
  return new Promise((resolve) => {
    setTimeout(() => {
      console.log('User name fetched');
      resolve('Williams');
    }, 1000);
  });
}
function UserAge() {
  return new Promise((resolve) => {
    setTimeout(() => {
      console.log('User age fetched');
      resolve(42);
    }, 1000);
  });
}
function UserCity() {
  return new Promise((resolve) => {
    setTimeout(() => {
      console.log('User city fetched');
      resolve('Switzerland');
    }, 1000);
  });
}
async function getUserDetails() {
  const name = await UserName();
  console.log('Name:', name);
  const age = await UserAge();
  console.log('Age:', age);
  const city = await UserCity();
  console.log('City:', city);
}
getUserDetails();
</script>
</body>
</html>
```

