

Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 1

Date Of Performance: 10/07/2025

Aim: Implementation of HTML Web development

THEORY:

What is HTML?

HTML (HyperText Markup Language) is the fundamental building block of the World Wide Web. It's a markup language used to:

- Create and structure content for web pages
- Define the meaning and structure of web content
- Enable the display of text, images, videos, and other multimedia

Basic Concepts

1. Markup Language

HTML is not a programming language but a markup language that uses tags to annotate text and other content for display in web browsers.

2. Elements and Tags

HTML documents are made up of elements, which are represented by tags:

- Opening tag: <tagname>
- Content: The actual content to be displayed
- Closing tag: </tagname>

Example: <p>This is a paragraph.</p>

3. Document Structure

A basic HTML document has this structure:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <!-- Content goes here -->
  </body>
</html>
```

Key Components

1. DOCTYPE Declaration

<!DOCTYPE html> defines the document type and HTML version (HTML5 in this case).

2. HTML Element

The <html> element is the root element that contains all other HTML elements.

3. Head Section

The <head> contains meta-information about the document:

- <title>: Sets the browser tab title
- <meta>: Provides metadata like character set, viewport settings
- <link>: Links to external resources like CSS
- <script>: Contains or links to JavaScript code

4. Body Section

The <body> contains all the visible content:

- Text elements (<h1> to <h6>, <p>, , etc.)
- Media elements (, <video>, <audio>)
- Structural elements (<div>, <section>, <header>, <footer>)
- Interactive elements (<a>, <button>, <form>)

HTML Principles

1. Semantic HTML: Using elements that convey meaning about the content structure (like <article>, <nav>, <footer>)
2. Accessibility: Writing HTML that's accessible to all users, including those with disabilities

3. Separation of Concerns: Keeping content (HTML), presentation (CSS), and behavior (JavaScript) separate
4. Progressive Enhancement: Building from basic content to more complex features

Evolution of HTML

- HTML 1.0 (1993): The first version
- HTML 2.0 (1995): Standardized features
- HTML 4.01 (1999): Added CSS support
- XHTML (2000): XML-based strict version
- HTML5 (2014): Current standard with multimedia, semantic elements, and APIs

CODE:

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet"
href="index.css" />
  </head>
  <body>
    <main>
      <h1>
        You have been invited to a
        <pre class="title">birthday
party</pre>
      </h1>
      <form action="">
        <label for="name">
          <span>Name :</span>
          <input name="name"
type="text" placeholder="Enter name" />
        </label>
        <label for="time">
          <span>Time :</span>
          <input name="time"
type="text" placeholder="Enter time" />

```

```
        </label>
        <label for="place">
          <span>Place :</span>
          <input name="place"
type="text" placeholder="Enter place" />
        </label>
        <section>
          <button>Clear</button>
          <button>Submit</button>
        </section>
      </form>
      <a href="gifts.html">Send Gift
    > </a>
    </main>
    
  </body>
</html>
```

gift.html

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

```

```

<title>My Wonderful Gifts</title>
<link rel="stylesheet"
href="index.css" />
</head>
<body>
  <main>
    <h1>🎁 Gifts Received So Far 🎁</h1>
    <ul>
      <li>🥭 Mangoes</li>
      <li>🕒 Watch</li>
      <li>🧢 Cap</li>
      <li>👕 T-Shirt</li>
    </ul>
    <a href="index.html"> <
      Back</a>
    </main>
    
  </body>
</html>

```

index.css

```

* {
  margin: 0;
  padding: 0;
}

body {
  font-family: system-
ui, -apple-system,
BlinkMacSystemFont, "Segoe
UI",
  Roboto, Oxygen,
Ubuntu, Cantarell, "Open
Sans", "Helvetica Neue",
  sans-serif;
  overflow-x: hidden;
  position: relative;
  display: flex;
  justify-content:
center;
  align-items: center;
  width: 100vw;
  height: 100vh;
}

h1,
h2,
h3,
h4,
h5,
h6 {
  font-weight: normal;
  text-align: center;
}

main {
  border-radius: 12px;
  display: flex;
  flex-direction:
column;
  gap: 2rem;
  color: #fff;
  padding: 2rem;
  background: rgba(255,
255, 255, 0.1);
  backdrop-filter:
blur(10px);
  -webkit-backdrop-
filter: blur(10px);
  border: 1px solid
rgba(255, 255, 255, 0.2);
  box-shadow: 0 4px 30px
rgba(0, 0, 0, 0.1);
}

form {
  display: flex;
  flex-direction:
column;
  gap: 1rem;
  width: inherit;
}

label {
  display: flex;
  flex-direction: row;
  align-items: center;
  gap: 0.5rem;
}

span {
  font-size: large;
  width: 20%;
}

input {
  width: 100%;
  border-radius: 5px;
  padding: 0.35rem;
  font-size: medium;
}

section {
  width: 100%;
  display: flex;
  gap: 1rem;
  justify-items: center;
  align-items: center;
}

button {
  width: 100%;
  border-radius: 5px;
  padding: 0.35rem;
  font-size: medium;
  cursor: pointer;
}

img {
  z-index: -1;
  width: 100vw;
  height: 100vh;
  position: absolute;
  object-fit: cover;
  inset: 0;
}

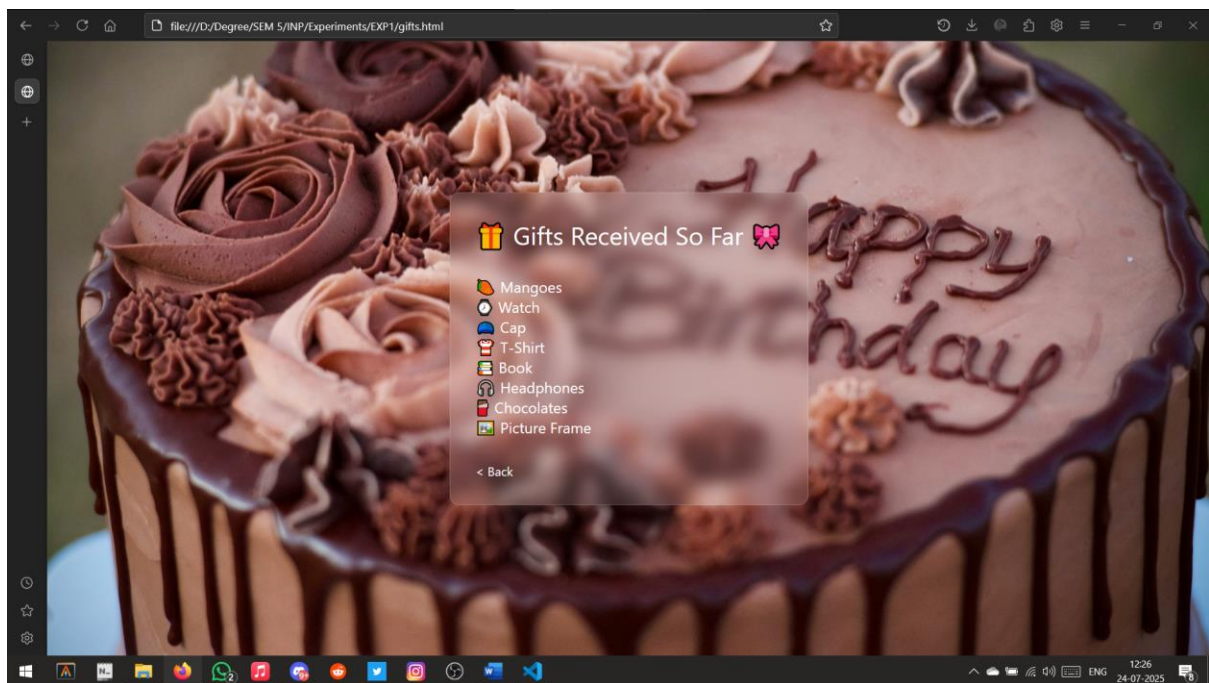
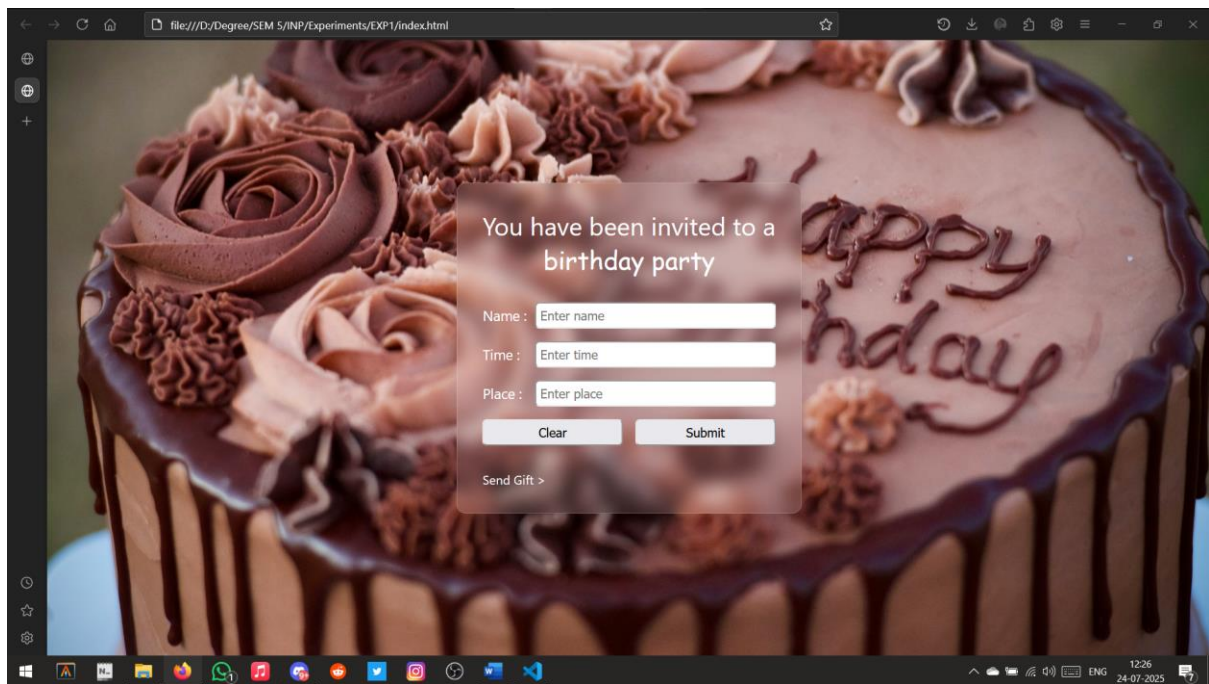
a {
  color: #fff;
  text-decoration: none;
}

li {
  list-style-type: none;
  font-size: larger;
}

.title {
  text-align: center;
  font-size: xx-large;
  font-family: cursive;
}

```

OUTPUT:



Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 2

Date Of Performance: 24/07/2025

Aim: Implementation of CSS

THEORY:

1. What is CSS?

- Styles HTML elements
- Controls layout, colors, fonts
- Makes websites responsive

2. Three Ways to Apply CSS

A. Inline CSS

- Added directly to HTML tags
- Highest priority
- Example: `<p style="color:red;">Red text</p>`

Pros:

- ✓ Quick to apply
- ✓ Overrides other styles

Cons:

- X Hard to maintain
- X Not reusable

B. Internal CSS

- Placed in `<style>` tag in `<head>`
- Applies to single page
- Example:

```
<style> p { color: blue; } </style>
```

Pros:

- ✓ Better than inline
- ✓ Good for small projects

Cons:

- X Not for multiple pages

C. External CSS

- Separate .css file
- Linked with `<link>` tag
- Best for large sites
- Example:

```
<link rel="stylesheet" href="style.css">
```

Pros:
✓ Easy maintenance
✓ Faster loading
✓ Reusable

1. Inline CSS (strongest)
2. Internal CSS
3. External CSS (weakest)

Cons:
X Extra file needed

Best Practices

- Use External CSS for big projects
- Use Inline only for quick fixes
- Internal for testing/small pages

3. Which One Wins?

Priority Order:

CODE:

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta
      charset="UTF-8" />
    <meta
      name="viewport"
      content="width=device-
        width, initial-scale=1.0"
      />
    <title>Login
      Form</title>
    <link
      href="https://
        cdn.jsdelivr.net/npm/boots
        trap@5.3.3/dist/css/bootst
        rap.min.css"
      rel="styleshee
        t"
      />
    <style>
      :root {
        --dark-bg:
          #212121;
        --darker-
          bg: #262626;
        --card-bg:
          #333333;
        --text-
          color: #ccc;
```

```
        color: #ccc;
        --primary-
          color: #555555;
        --hover-
          color: #555555;
        body {
          min-
            height: 100vh;
          display:
            flex;
          flex-
            direction: column;
          background
            -color: var(--dark-bg);
          color:
            var(--text-color);
          .navbar {
            background
              -color: var(--darker-bg)
              !important;
            .navbar-brand,
              .nav-link {
                color:
                  var(--text-color)
                  !important;
              }
            .nav-
              link:hover {
```

```
              color:
                var(--primary-color)
                !important;
            }
            .navbar-
              toggler {
                border-
                  color: rgba(255, 255, 255,
                    0.1);
            }
            .navbar-
              toggler-icon {
                background
                  -image:
                    url("data:image/svg+xml,%3
                      csvg
                      xmlns='http://www.w3.org/2
                      000/svg' viewBox='0 0 30
                      30'%3e%3cpath
                      stroke='rgba%28255, 255,
                      255, 0.75%29' stroke-
                      linecap='round' stroke-
                      miterlimit='10' stroke-
                      width='2' d='M4 7h22M4
                      15h22M4
                      23h22'/%3e%3c/svg%3e");
            }
            .form-
              container {
                flex: 1;
                display:
                  flex;
                align-
                  items: center;
```

```

        justify-
content: center;
        padding:
2rem;
    }

    .login-form {
        width:
100%;
        max-width:
600px;
        padding:
2rem;
        border-
radius: 10px;
        box-
shadow: 0 10px 20px
        rgba(0, 0, 0, 0.3);
        background
-color: var(--card-bg);
        border:
1px solid rgba(255, 255,
255, 0.1);
    }

    .login-title {
        text-
align: center;
        margin-
bottom: 1.5rem;
        color:
var(--primary-color);
        font-
weight: 600;
    }

    .form-control,
    .form-select {
        background
-color: rgba(0, 0, 0,
0.2);
        border-
color: rgba(255, 255, 255,
0.1);
        color:
var(--text-color);
    }

    .form-
control:focus,
    .form-
select:focus {
        background
-color: rgba(0, 0, 0,
0.3);

```

```

        border-
color: var(--primary-
color);
        color:
var(--text-color);
        box-
shadow: 0 0 0 0.25rem
        rgba(76, 201, 240, 0.25);
    }

    .form-label {
        color:
var(--text-color);
    }

    .btn-primary {
        background
-color: var(--primary-
color);
        border-
color: var(--primary-
color);
        color:
#111;
    }

    .btn-
primary:hover {
        background
-color: var(--hover-
color);
        border-
color: var(--hover-color);
    }

    .form-check-
input {
        background
-color: rgba(0, 0, 0,
0.2);
        border-
color: rgba(255, 255, 255,
0.1);
    }

    .form-check-
input:checked {
        background
-color: var(--primary-
color);
        border-
color: var(--primary-
color);
    }
}
</style>
</head>
<body>

```

```

<nav class="navbar
navbar-expand-lg navbar-
dark">
    <div
class="container-fluid">
        <a
class="navbar-brand"
href="#">INPEXP2</a>
        <button
class=
"navbar-toggler"
type="
button"
data-
bs-toggle="collapse"
data-
bs-target="#navbarNav"
aria-
controls="navbarNav"
aria-
expanded="false"
aria-
label="Toggle navigation"
>
            <span
class="navbar-toggler-
icon"></span>
        </button>
        <div
class="collapse navbar-
collapse" id="navbarNav">
            <ul
class="navbar-nav">
                <li
class="nav-item">
                    <a
class="nav-link
active"
aria-current="page"
href="#"
>Home</a>
                </li>
            </ul>
            <li
class="nav-item">
                <a class="nav-link

```



```

active"
href="#">Features</a>
    </li>
    <li>
        <a class="nav-link
active" href="#">About</a>
    </li>
    <li>
        <a class="nav-link
active" href="#">Login</a>
    </li>
    </ul>
</div>
<div class="form-
container">
    <div
class="login-form">
        <h2
class="login-
title">Register</h2>
        <form
class="row g-3">
            <div
class="col-md-6">
                <label
for="inputEmail4"
class="form-label">
                    Email</label>
                <input
type="email"
class="form-control"
id="inputEmail4"
/>
            </div>
            <div
class="col-md-6">
                <label
for="inputPassword4"
class="form-label">
                    Password</label>
                <input
type="password"
class="form-control"
id="inputPassword4"
/>
            </div>
            <div
class="col-12">
                <label
for="inputAddress"
class="form-label">
                    Address</label>
                <input
type="text"
class="form-control"
id="inputAddress"
placeholder="1234 Main
St"
/>
            </div>
            <div
class="col-12">
                <label
for="inputAddress2"
class="form-label">
                    Address 2</label>
                <input
type="text"
class="form-control"
id="inputAddress2"
/>
            </div>
        </form>
    </div>
    <div
class="col-md-4">
        <label
for="inputState"
class="form-
label">State</label>
        <select
id="inputState"
class="form-select">
            <option
selected>Choose...</option>
            <option>...</option>
        </select>
    </div>
    <div
class="col-md-2">
        <label
for="inputZip"
class="form-
label">Zip</label>
        <input
type="text"
class="form-control"
id="inputZip" />
    </div>
</div>
<div
class="col-12">
    <div class="form-check">

```

```

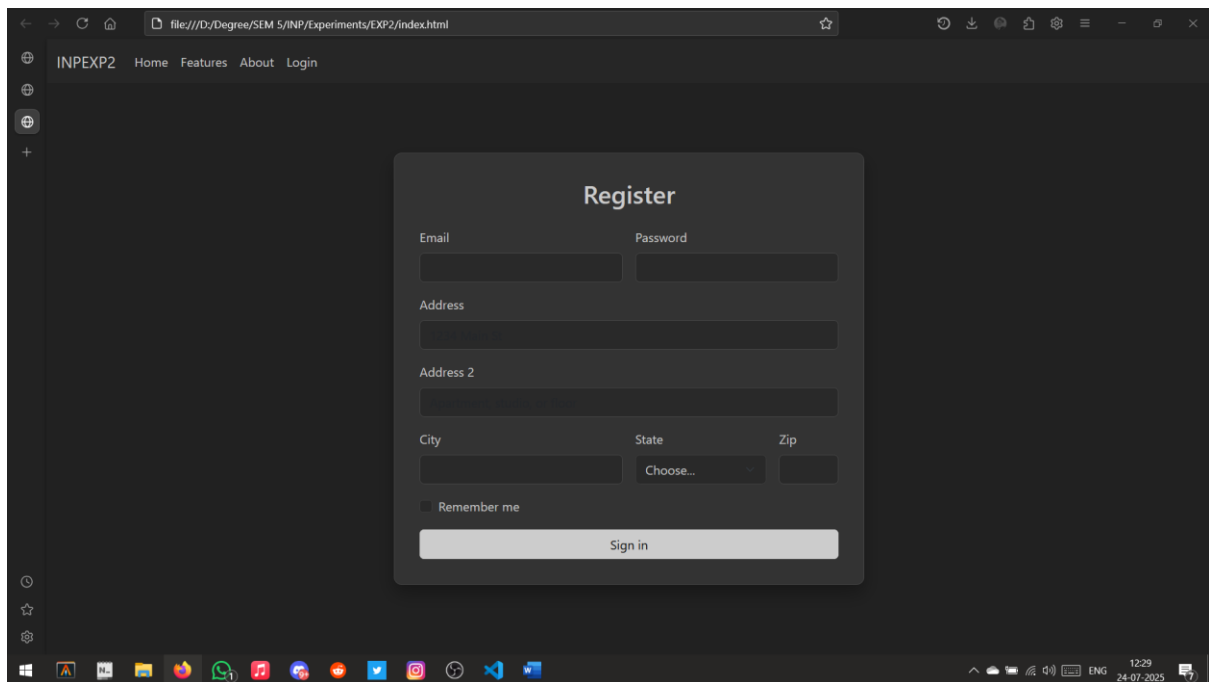
<input                                Remember me                                button>                                </
                                </div>
                                </form>
                                </div>
                                </div>
                                <script
                                src="https://cdn.jsdelivrivr.
                                net/npm/bootstrap@5.3.3/di
                                st/js/bootstrap.bundle.min
                                .js"></script>
                                </body>
                                </html>

class="form-check-
input"
                                </label>
                                </div>
                                </div>
                                <div
                                class="col-12">
                                <b
                                utton type="submit"
                                class="btn btn-primary w-
                                100">
                                Sign in

                                type="checkbox"
                                id="gridCheck"
                                />
                                <label class="form-
                                check-label"
                                for="gridCheck">

```

OUTPUT:



Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 3

Date Of Performance: 31/07/2025

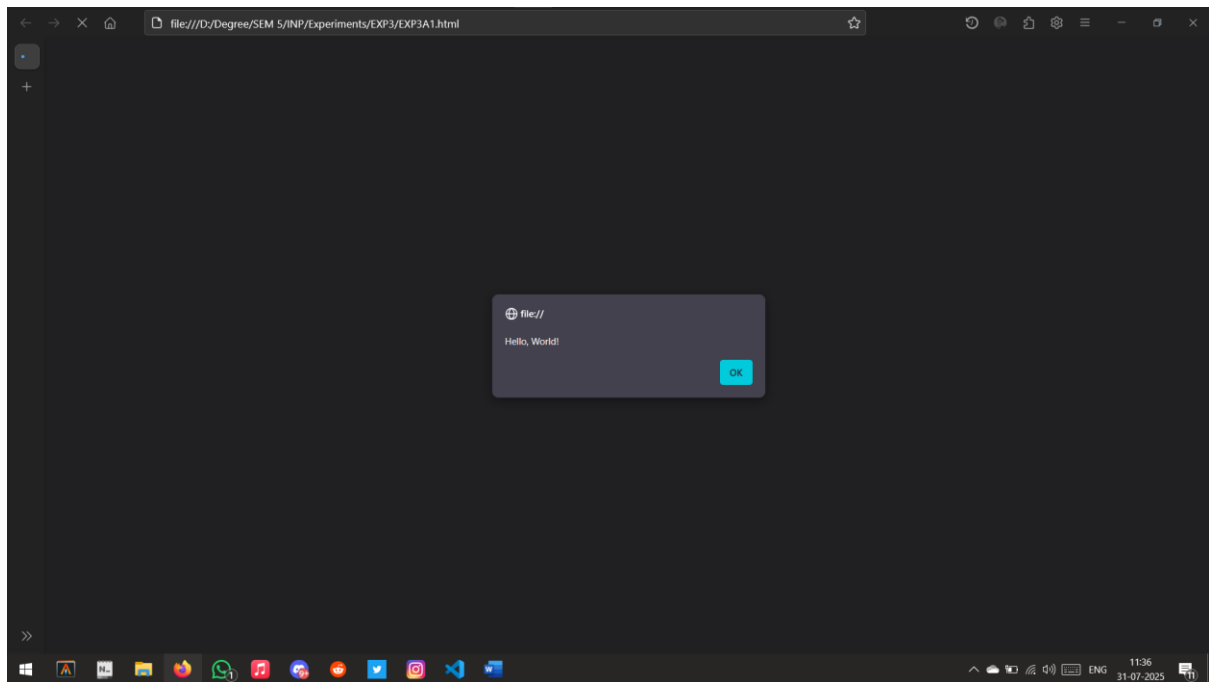
Aim: Implementation of Frontend Web Application
Development Bootstrap.

Q) Implement hello world example in java script. Run on browser and console using node

CODE (Browser):

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <script>
      console.log("Hello, World! (from Firefox)");
      alert("Hello, World!");
    </script>
  </body>
</html>
```

OUTPUT:



CODE (Console):

```
console.log("Hello, World! (from Node)");
```

OUTPUT:

```
Abdurrahman Qureshi@DESKTOP-H2RV5MQ MINGW64 /d/Degree/SEM 5/INP/Experiments/EXP3 (master)
$ node EXP3A2.js
Hello, World! (from Node)
```

Q) Implement DOM manipulation for change in color of text to red

CODE:

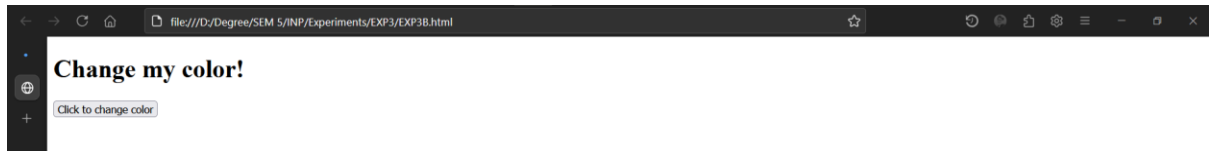
```
<!DOCTYPE html>
<html>
  <head>
    <title>Change to Red</title>
  </head>
  <body>
    <h1 id="text">Change my color!</h1>
```

```
    <button
      onclick="changeColor()">Click to change
      color</button>
    <script>
      const changeColor = () => {
        const heading =
          document.getElementById("text");
        heading.style.color =
          "red";
```

```
};  
</script>
```

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

OUTPUT:



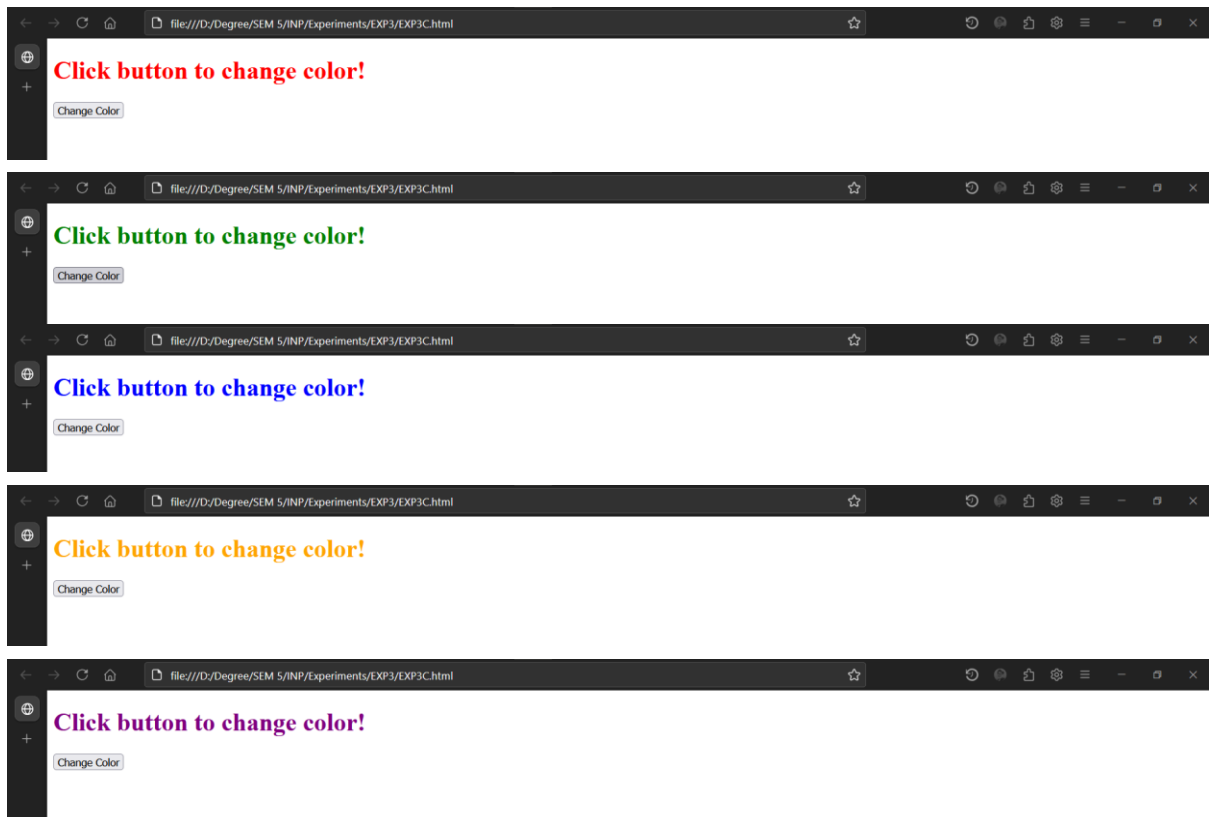
Q) Implement above using change in different colors by clicking button using Java script

CODE:

```
<!DOCTYPE html>                                     let index = 0;  
  
<html>                                                function changeColor() {  
  <head>                                              const text =  
    <title>Color Changer</title>                    document.getElementById("text");  
  </head>                                              text.style.color =  
  <body>                                              colors[index];  
    <h1 id="text">Click button to change color!</h1>  index = (index + 1) %  
    <button                                           colors.length;  
      onclick="changeColor()">Change                }  
      Color</button>                                </script>  
  <script>                                           </body>  
    const colors = ["red", "green",                 </html>  
    "blue", "orange", "purple"];                    </script>
```

OUTPUT:





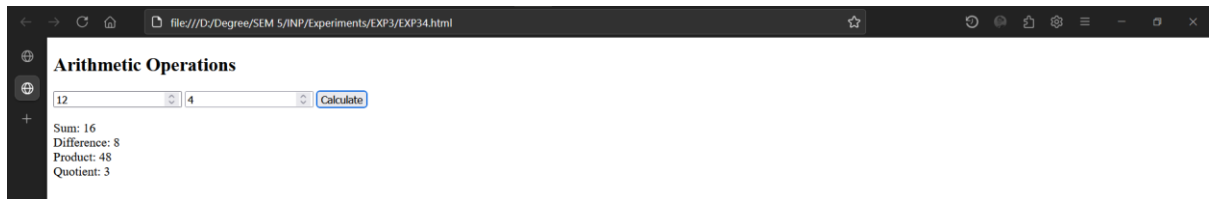
Q) Implement arithmetic operations using Java script both on browser and console

CODE (Browser):

```
<!DOCTYPE html>
<html>
  <head>
    <title>Arithmetic</title>
  </head>
  <body>
    <h2>Arithmetic Operations</h2>
    <input id="num1" type="number"
placeholder="Enter number 1" />
    <input id="num2" type="number"
placeholder="Enter number 2" />
    <button
onclick="calculate()">Calculate</button>
    <p id="result"></p>
    <script>
      function calculate() {
```

```
        const n1 =
parseFloat(document.getElementById("num1").
value);
        const n2 =
parseFloat(document.getElementById("num2").
value);
        const result = `
          Sum: ${n1 + n2} <br>
          Difference: ${n1 - n2} <br>
          Product: ${n1 * n2} <br>
          Quotient: ${n1 / n2}
        `;
        document.getElementById("re
sult").innerHTML = result;
      }
    </script>
  </body>
</html>
```

OUTPUT:

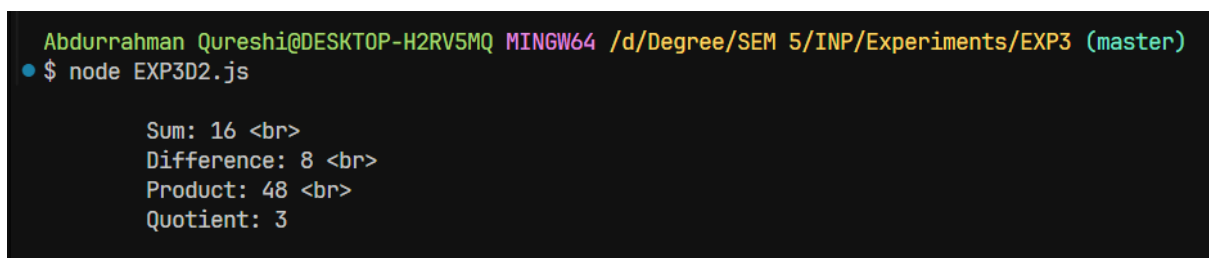


CODE (Console):

```
function calculate() {  
  const n1 = 12;  
  const n2 = 4;  
  const result = `  
    Sum: ${n1 + n2} <br>  
    Difference: ${n1 - n2} <br>  
    Product: ${n1 * n2} <br>  
    Quotient: ${n1 / n2}  
  `;  
  console.log(result);  
}
```

```
calculate();
```

OUTPUT:



Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 4

Date Of Performance: 14/08/2025

Aim: Implementation of Basic Java Script ES5.

Q) Implement a grading system with multiple inputs

CODE:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0" />
    <title>Grade Calculator</title>
    <link
      href="https://cdn.jsdelivr.net/
npm/tailwindcss@2.2.19/dist/tailwind.min.cs
s"
      rel="stylesheet"
    />
  </head>
  <body class="min-h-screen flex items-
center justify-center p-4">
    <div class="rounded-lg shadow-md p-
6 w-full max-w-md">
      <h1 class="text-2xl font-bold
text-center mb-6">
        Grade Calculator
      </h1>
      <div class="space-y-4">
        <div class="flex flex-col
gap-2">
          <label for="score"
class="block text-sm font-medium mb-1"
            >Enter your score
            (0-100)</label>
          <input
            type="number"
            id="scoreA"
            min="0"
            max="100"
            class="w-full px-4
py-2 border border-gray-300 rounded-md
focus:outline-none"
            placeholder="e.g.
85"
          />
          <input
            type="number"
            id="scoreB"
            min="0"
            max="100"
            class="w-full px-4
py-2 border border-gray-300 rounded-md
focus:outline-none"
          />
        </div>
      </div>
    </div>
  </body>
</html>
```



```

placeholder="e.g.
85"
    />
    <input
        type="number"
        id="scoreC"
        min="0"
        max="100"
        class="w-full px-4
py-2 border border-gray-300 rounded-md
focus:outline-none"
        placeholder="e.g.
85"
    />
</div>

<button
    onclick="calculateGrade
()"
    class="w-full font-
medium py-2 px-4 rounded-md transition
duration-200"
    >
        Calculate Grade
</button>

<div id="result"
class="hidden mt-6 p-4 rounded-md text-
center">
    <h2 class="text-lg
font-semibold mb-2">Your Grade:</h2>
    <div id="grade"
class="text-4xl font-bold"></div>
    <p id="message"
class="mt-2 text-gray-600"></p>
</div>
</div>

<script>
    function calculateGrade() {
        const scoreInputA =
document.getElementById("scoreA");
        const scoreInputB =
document.getElementById("scoreB");
        const scoreInputC =
document.getElementById("scoreC");
        const resultDiv =
document.getElementById("result");

```

```

        const gradeDisplay =
document.getElementById("grade");
        const messageDisplay =
document.getElementById("message");

        if (
            !scoreInputA.value ||
            scoreInputA.value < 0
            ||
            scoreInputA.value > 100
            ||
            !scoreInputB.value ||
            scoreInputB.value < 0
            ||
            scoreInputB.value > 100
            ||
            !scoreInputC.value ||
            scoreInputC.value < 0
            ||
            scoreInputC.value > 100
        ) {
            alert("Please enter a
valid score between 0 and 100");
            return;
        }

        const score =
            (parseInt(scoreInputA.v
            alue) +
                parseInt(scoreInput
            B.value) +
                parseInt(scoreInput
            C.value)) /
            3;
        console.log(score);
        let grade, message;

        if (score >= 90) {
            grade = "A";
            message = "Excellent
work!";
        } else if (score >= 80) {
            grade = "B";
            message = "Good job!";
        } else if (score >= 70) {
            grade = "C";
            message = "You
passed!";
        } else if (score >= 60) {

```

```

        grade = "D";
        message = "You barely
passed";
    } else {
        grade = "F";
        message = "You failed.
Try harder next time.";
    }

    gradeDisplay.textContent =
grade;
    messageDisplay.textContent
= message;

    resultDiv.classList.remove(
        "bg-green-100",
        "text-green-800",
        "bg-blue-100",
        "text-blue-800",
        "bg-yellow-100",
        "text-yellow-800",
        "bg-orange-100",
        "text-orange-800",
        "bg-red-100",
        "text-red-800"
    );

    if (grade === "A") {
        resultDiv.classList.add
("bg-green-100", "text-green-800");
    } else if (grade === "B") {
        resultDiv.classList.add
("bg-blue-100", "text-blue-800");
    } else if (grade === "C") {
        resultDiv.classList.add
("bg-yellow-100", "text-yellow-800");
    } else if (grade === "D") {
        resultDiv.classList.add
("bg-orange-100", "text-orange-800");
    } else {
        resultDiv.classList.add
("bg-red-100", "text-red-800");
    }

    resultDiv.classList.remove(
        "hidden");
}
</script>
</body>
</html>

```

OUTPUT:

Grade Calculator

Enter your score (0-100)

65

98

82

Calculate Grade

Your Grade:

B

Good job!

Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 5

Date Of Performance: 14/08/2025

Aim: Implementation of Java Script ES6.

Q) Implement DOM manipulation to demonstrate a login form

CODE:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0" />
    <title>Login</title>
    <link
      href="https://cdn.jsdelivr.net/
npm/tailwindcss@2.2.19/dist/tailwind.min.cs
s"
      rel="stylesheet"
    />
  </head>
  <body
    class="w-screen h-screen flex flex-
row justify-center items-center bg-gray-
100"
  >
    <form
      class="flex flex-col gap-4 w-
full max-w-md p-6 bg-white rounded-lg
shadow-md"
    >
```

```
      <h1 class="text-2xl font-bold
text-center text-gray-800">Login</h1>

      <div class="flex flex-col gap-
1">
        <label for="email"
class="text-sm font-medium text-gray-700"
          >Email</label>
        >
        <input
          id="email"
          name="email"
          type="email"
          required
          class="border border-
gray-300 p-2 rounded-md focus:outline-none
focus:ring-2 focus:ring-blue-500"
          placeholder="Enter your
email"
        />
      </div>

      <div class="flex flex-col gap-
1">
        <label for="password"
class="text-sm font-medium text-gray-700"
```

```

        >Password</label>
    >
    <input
        id="password"
        name="password"
        type="password"
        required
        class="border border-
gray-300 p-2 rounded-md focus:outline-none
focus:ring-2 focus:ring-blue-500"
        placeholder="Enter your
password"
    />
</div>

<button
    type="submit"
    class="bg-blue-600
hover:bg-blue-700 text-white font-medium
py-2 px-4 rounded-md transition-colors
duration-200 focus:outline-none focus:ring-
2 focus:ring-blue-500 focus:ring-offset-2"
    >
    Sign In
</button>

<!-- <p class="text-center
text-sm text-gray-600 mt-4">
    Demo credentials:<br />
    Email:
scythethroughtheheart05@gmail.com<br />
    Password: iLoveGymMommies05
</p> -->
</form>

```

```

<script>
    document
        .querySelector("form")
        .addEventListener("submit",
function (e) {
            e.preventDefault();

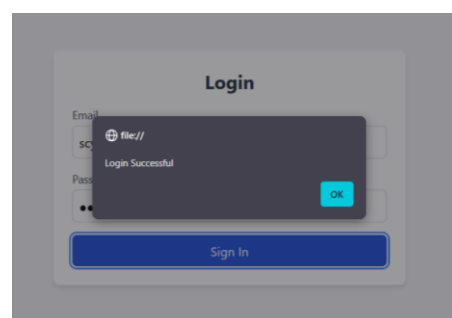
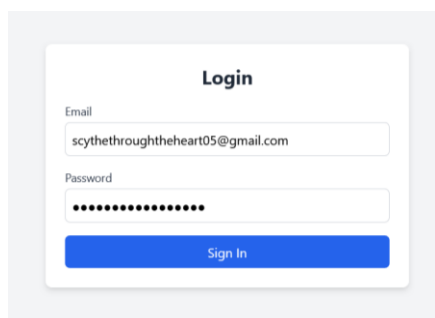
            const email =
document.getElementById("email").value;
            const password =
document.getElementById("password").value;

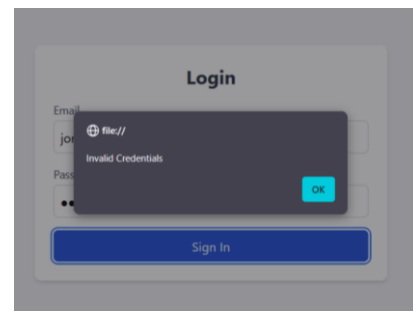
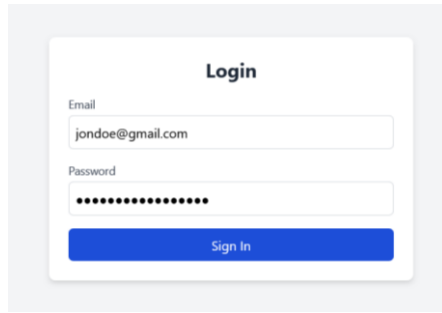
            if (email.length === 0
|| password.length === 0) {
                alert("Please fill
in all fields");
                return;
            }

            if (
                email ===
"scythethroughtheheart05@gmail.com" &&
                password ===
"iLoveGymMommies05"
            ) {
                alert("Login
Successful");
            } else {
                alert("Invalid
Credentials");
            }
        });
</script>
</body>
</html>

```

OUTPUT:





Q) Implement an age calculator

CODE:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0" />
    <title>Age Calculator</title>
    <link
      href="https://cdn.jsdelivr.net/
npm/tailwindcss@2.2.19/dist/tailwind.min.cs
s"
      rel="stylesheet"
    />
  </head>
  <body class="w-screen h-screen flex
flex-row justify-center items-center">
    <form
      class="flex flex-col gap-4 w-
full max-w-md bg-white p-8 rounded-lg
shadow-md"
    >
      <h1 class="text-3xl font-bold
text-center text-gray-800 mb-4">
        Age Calculator
      </h1>
      <div class="flex flex-col gap-
1">
        <label for="dob"
class="text-gray-700">Date of Birth</label>
        <input
          id="dob"
          type="date"

```

```
      class="border border-
gray-300 p-2 rounded-md focus:outline-none"
        max=""
      />
    </div>
    <button
      type="button"
      onclick="calculateAge()"
      class="mt-4 text-black
font-medium py-2 px-4 rounded-md transition
duration-200"
    >
      Calculate Age
    </button>
    <div id="result" class="mt-6 p-
4 rounded-md hidden">
      <h2 class="text-xl font-
semibold">Your Age</h2>
      <div class="mt-2 grid grid-
cols-3 gap-4 text-center">
        <div>
          <p class="text-4xl
font-bold" id="years">0</p>
          <p class="text-
black">Years</p>
        </div>
        <div>
          <p class="text-4xl
font-bold" id="months">0</p>
          <p class="text-
black">Months</p>
        </div>
      </div>
    </div>
```

```

        <div>
            <p class="text-4xl
font-bold" id="days">0</p>
            <p class="text-
black">Days</p>
        </div>
    </div>
</div>
</form>
<script>
    document.getElementById("dob").
max = new Date()
        .toISOString()
        .split("T")[0];
    function calculateAge() {
        const dobInput =
document.getElementById("dob");
        const resultDiv =
document.getElementById("result");

        if (!dobInput.value) {
            alert("Please select
your date of birth");
            return;
        }
        const dob = new
Date(dobInput.value);
        const today = new Date();
        let years =
today.getFullYear() - dob.getFullYear();
        let months =
today.getMonth() - dob.getMonth();

        let days = today.getDate()
- dob.getDate();
        if (days < 0) {
            months--;
            const lastMonth = new
Date(
                today.getFullYear()
,
                today.getMonth(),
                0
            );
            days +=
lastMonth.getDate();
        }
        if (months < 0) {
            years--;
            months += 12;
        }
        document.getElementById("ye
ars").textContent = years;
        document.getElementById("mo
nths").textContent = months;
        document.getElementById("da
ys").textContent = days;

        resultDiv.classList.remove(
"hidden");
    }
</script>
</body>
</html>

```

OUTPUT:

Age Calculator

Date of Birth

Your Age

20
Years

7
Months

24
Days

Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 6

Date Of Performance: 14/08/2025

Aim: Implementation of Asynchronous Java Script using
callback, promise, fetch etc.

Q) Implement the use of fetch via any API

CODE:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
content="width=device-width, initial-
scale=1.0" />
    <title>Waifu.pics Demo</title>
    <style>
      body {
        font-family: Arial, sans-
serif;
        max-width: 800px;
        margin: 0 auto;
        padding: 20px;
        text-align: center;
      }
      .waifu-container {
        margin: 30px 0;
      }
      #waifu-image {
        max-width: 100%;
        max-height: 500px;
        border-radius: 8px;
        box-shadow: 0 4px 8px
        rgba(0, 0, 0, 0.1);
      }
      button {
        background-color: #ff6b81;
        color: white;
        border: none;
        padding: 10px 20px;
        font-size: 16px;
        border-radius: 5px;
        cursor: pointer;
        margin: 5px;
        transition: background-
color 0.3s;
      }
      button:hover {
        background-color: #ff4757;
      }
      .category-buttons {
        margin-bottom: 20px;
      }
```

```

        </style>
    </head>
    <body>
        <h1>Waifu.pics Demo</h1>
        <p>Click a button to get a random
        waifu image!</p>

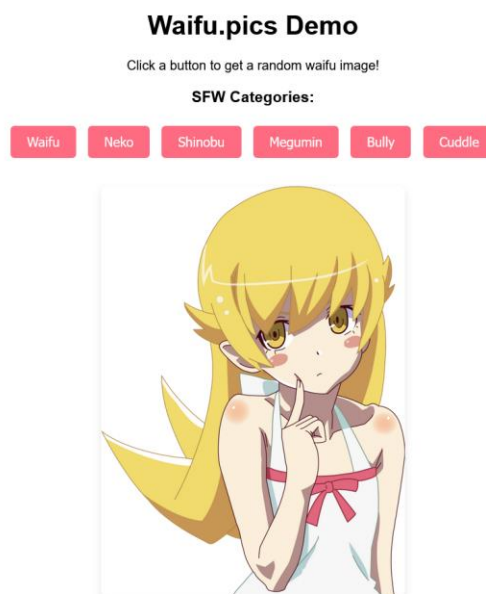
        <div class="category-buttons">
            <h3>SFW Categories:</h3>
            <button
            onclick="getWaifu('waifu')">Waifu</button>
            <button
            onclick="getWaifu('neko')">Neko</button>
            <button
            onclick="getWaifu('shinobu')">Shinobu</butt
            on>
            <button
            onclick="getWaifu('megumin')">Megumin</butt
            on>
            <button
            onclick="getWaifu('bully')">Bully</button>
            <button
            onclick="getWaifu('cuddle')">Cuddle</button
            >
        </div>
        <div class="waifu-container">

            <img id="waifu-image" src=""
            alt="Random waifu will appear here" />
        </div>
        <script>
            function getWaifu(category) {
                fetch(`https://api.waifu.pics/sfw/${category}`)
                    .then((response) =>
                    response.json())
                    .then((data) => {
                        document.getElement
                        ById("waifu-image").src = data.url;
                    })
                    .catch((error) => {
                        console.error("Erro
                        r:", error);

                        alert("Failed to
                        fetch waifu image. Please try again.");
                    });
            }
            window.onload = function () {
                getWaifu("waifu");
            };
        </script>
    </body>
</html>

```

OUTPUT:



Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 7

Date Of Performance: 11/09/2025

Aim: Implementation of Hello World Application using React
CDN/ CRA/ Vite.

```
Abdurrahman Qureshi@DESKTOP-H2RV5MQ MINGW64 ~/Desktop
$ npx create-react-app exp7

Creating a new React app in C:\Users\Abdurrahman Qureshi\Desktop\exp7.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1324 packages in 1m

271 packages are looking for funding
  run `npm fund` for details

Initialized a git repository.

Installing template dependencies using npm...

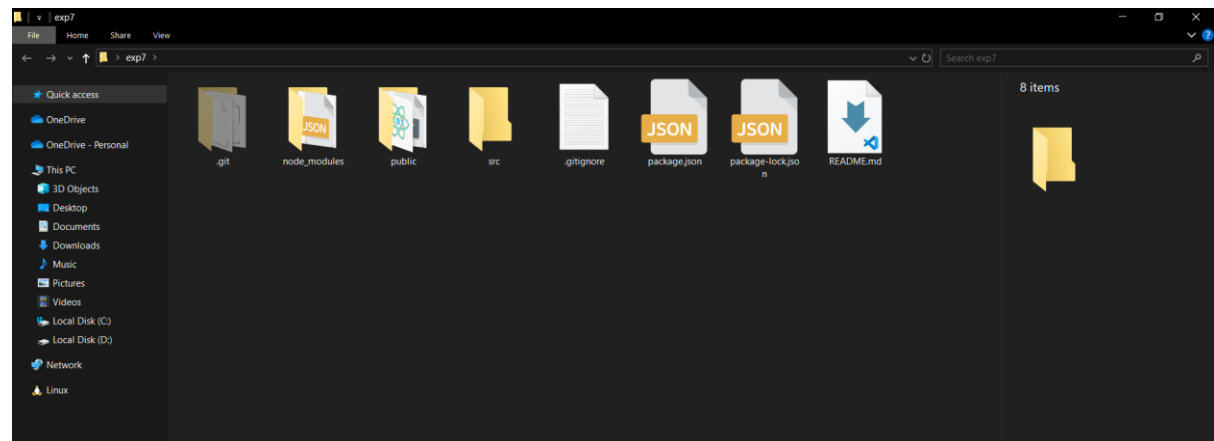
added 17 packages, and changed 1 package in 11s

271 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...

removed 1 package, and audited 1341 packages in 7s

271 packages are looking for funding
  run `npm fund` for details

9 vulnerabilities (3 moderate, 6 high)
```



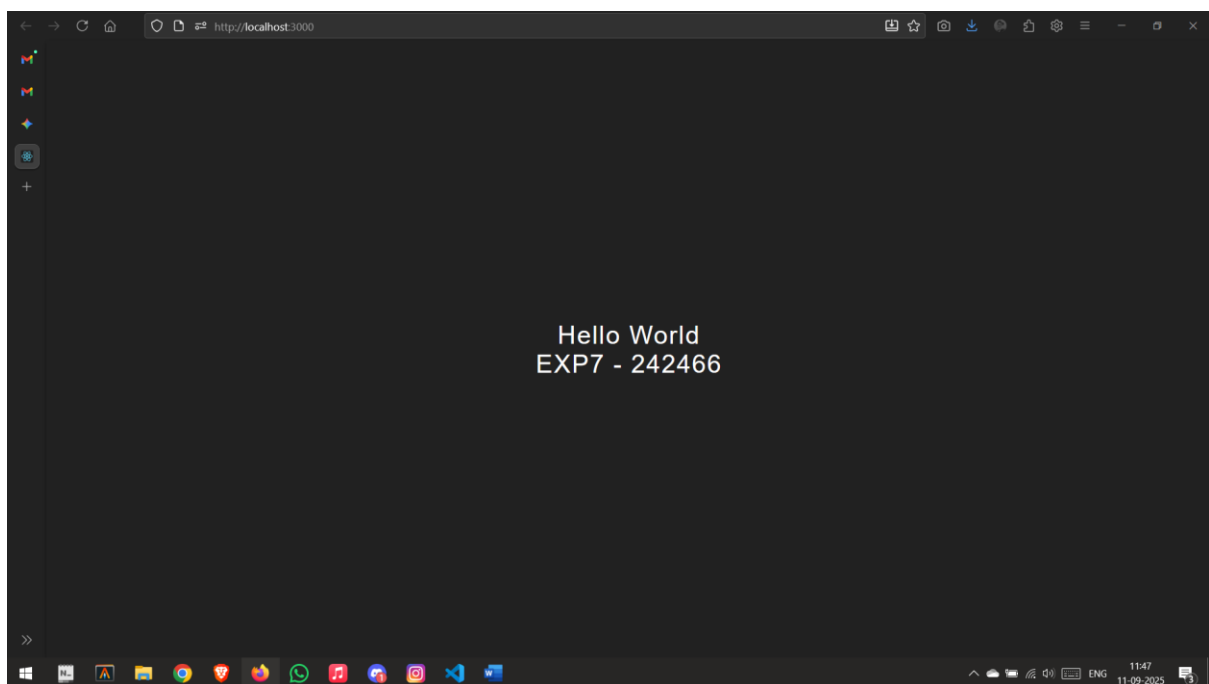
CODE:

```
import './App.css';

function App() {
  return (
    <>
      Hello World
      <br />
      EXP7 - 242466
    </>
  );
}

export default App;
```

OUTPUT:



Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 8

Date Of Performance: 18/09/2025

Aim: Implementation of React Functional Components and prop using Vite.

Q) Create functional components to render h1 and p tags

CODE (Quote):

```
export default function Hero() {
  return (
    <div className="h-screen w-screen flex justify-center items-center flex-col">
      <h1 className="text-5xl">"Revenge is fool's game"</h1>
      <p className="text-3xl">
        - Arthur Morgan
      </p>
    </div>
  );
}
```

CODE (Rendering):

```
import Hero from "./components/Hero";

export default function Home() {
  return (
    <>
      <Hero />
    </>
  );
}
```

OUTPUT:



Q) Create at least three functions with h1,p,li and image and render them

CODE (Gang):

```
"use client";

import Image from "next/image";
import IMG from "../../public/img/Red Dead Redemption Dutch.jpg";

export default function Gang() {
  const gangMembers = [
    { name: "Dutch van der Linde", role: "Gang leader" },
    { name: "Arthur Morgan", role: "Senior enforcer and lead debt collector" },
    { name: "John Marston", role: "Gunslinger and former delinquent" },
    { name: "Hosea Matthews", role: "Master con artist and co-founder" },
    { name: "Susan Grimshaw", role: "Camp overseer and disciplinarian" },
    { name: "Micah Bell", role: "Gunslinger and recent recruit" },
    { name: "Bill Williamson", role: "Enforcer and former soldier" },
    { name: "Javier Escuella", role: "Expert tracker and revolutionary" },
    { name: "Charles Smith", role: "Hunter and skilled warrior" },
    { name: "Lenny Summers", role: "Young protégé and intelligent operator" },
    { name: "Sean MacGuire", role: "Irish gunslinger" },
    { name: "Karen Jones", role: "Con artist and drinker" },
    { name: "Mary-Beth Gaskill", role: "Bookworm and pickpocket" },
    { name: "Tilly Jackson", role: "Former street gang member" },
    { name: "Leopold Strauss", role: "Loan shark and financier" },
    { name: "Josiah Trelawny", role: "Mysterious con artist" },
    { name: "Reverend Swanson", role: "Opium-addicted preacher" },
    { name: "Uncle", role: "Old hanger-on and self-proclaimed cancer survivor" },
    { name: "Molly O'Shea", role: "Dutch's romantic interest" },
    { name: "Abigail Roberts", role: "John Marston's love interest" },
    { name: "Jack Marston", role: "Abigail and John's young son" },
    { name: "Sadie Adler", role: "Recent widow turned fierce fighter" },
    { name: "Kieran Duffy", role: "Former O'Driscoll turned loyal member" },
    { name: "Simon Pearson", role: "Camp cook and butcher" }
  ];

  return (
    <div className="relative h-screen w-screen overflow-hidden">
      <Image
        src={IMG}
        alt="Red Dead Redemption 2"
        className="w-full h-full object-cover absolute top-0 left-0 z-10"
        width={1920}
        height={1080}
        priority
      />
    </div>
  );
}
```

```

    />
    <div className="relative z-20
h-full w-full flex flex-col items-center
justify-center bg-opacity-60 p-4">
      <h1 className="text-5xl
md:text-6xl font-bold text-amber-400 mb-2
font-rye drop-shadow-lg">
        Van der Linde Gang
      </h1>
      <p className="text-2xl
md:text-3xl text-amber-200 mb-6 font-
special">
        Members
      </p>

      <div className="grid grid-
cols-1 sm:grid-cols-2 md:grid-cols-3
lg:grid-cols-4 gap-4 w-full max-w-6xl px-4
overflow-y-auto py-4">
        {gangMembers.map((membe
r, index) => (

```

CODE (Rendering):

```

import Gang from "./components/Gang";

export default function Home() {
  return (
    <>
      <Gang />
    </>
  );
}

```

OUTPUT:



Q) Create a functional component with props

CODE (Animals):

```
"use client";

import Image from "next/image";
import IMG from "../../public/img/Red Dead Redemption Arthur (9).jpg";

interface Animal {
  name: string;
  location: string;
  species: string;
  desc: string;
}

export default function LegendaryAnimals({
  animals }: { animals: Animal[] }) {
  return (
    <div className="relative h-screen w-screen overflow-hidden">
      <Image
        src={IMG}
        alt="Red Dead Redemption 2"
        className="w-full h-full object-cover absolute top-0 left-0 z-10"
        width={1920}
        height={1080}
        priority
      />

      <div className="relative z-20 h-full w-full flex flex-col items-center justify-center bg-opacity-60 p-4">
        <h1 className="text-5xl md:text-6xl font-bold text-amber-900 mb-2 font-rye drop-shadow-lg">
          Legendary Animals
        </h1>
      </div>
    </div>
  );
}
```

```
    <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-4 w-full max-w-6xl px-4 py-4">
      {animals.map((animal, index) => (
        <div
          key={index}
          className="backdrop-blur-2xl bg-opacity-70 rounded-lg p-3 border transition-colors"
        >
          <h3
            className="text-xl font-bold text-amber-900">{animal.name}</h3>
          <p
            className="text-amber-900 text-sm mt-1">
            <span
              className="font-semibold">Location:</span>
            {animal.location}
          </p>
          <p
            className="text-amber-900 text-sm">
            <span
              className="font-semibold">Species:</span>
            {animal.species}
          </p>
          <p
            className="text-amber-900 text-xs mt-2">{animal.desc}</p>
        </div>
      ))}
    </div>
  );
}
```

CODE (Rendering):

```
import Animals from
"@/app/components/Animals";

export default function Home() {
  return (
    <>
      <Animals animals={animals} />
    </>
  );
}

const animals = [
  {
    name: "Legendary Bear",
    location: "Grizzlies East, north of O'Creagh's Run",
    species: "Grizzly Bear",
    desc: "A massive, scarred bear with unmatched strength and aggression."
  },
  {
    name: "Legendary Beaver",
    location: "Elysian Pool, north of Van Horn",
    species: "Beaver",
    desc: "A rare beaver with dark fur, skilled at evading predators."
  },
  {
    name: "Legendary Boar",
    location: "Bluewater Marsh, northeast of Lagras",
    species: "Boar",
    desc: "An unusually large boar with white fur, extremely aggressive."
  }
]
```

```

    },
    {
      name: "Legendary Buck",
      location: "West Elizabeth, near Big Valley",
      species: "Buck",
      desc: "A majestic buck with a white coat, agile and swift."
    },
    {
      name: "Legendary Cougar",
      location: "Gaptooth Ridge, west of Tumbleweed",
      species: "Cougar",
      desc: "Known as the Ghost Cat, stealthy and deadly."
    },
    {
      name: "Legendary Coyote",
      location: "Scarlett Meadows, southeast of Rhodes",
      species: "Coyote",
      desc: "A cunning coyote with a distinctive pale coat."
    },
    {
      name: "Legendary Elk",
      location: "Cumberland Forest, east of Bacchus Station",
      species: "Elk",
      desc: "A giant elk with huge antlers and white fur markings."
    },
    {
      name: "Legendary Fox",
      location: "Mattock Pond, north of Rhodes",
      species: "Fox",
      desc: "A rare fox with silver fur, quick and elusive."
    },
    {
      name: "Legendary Moose",
      location: "Roanoke Ridge, northeast of Annesburg",
      species: "Moose",

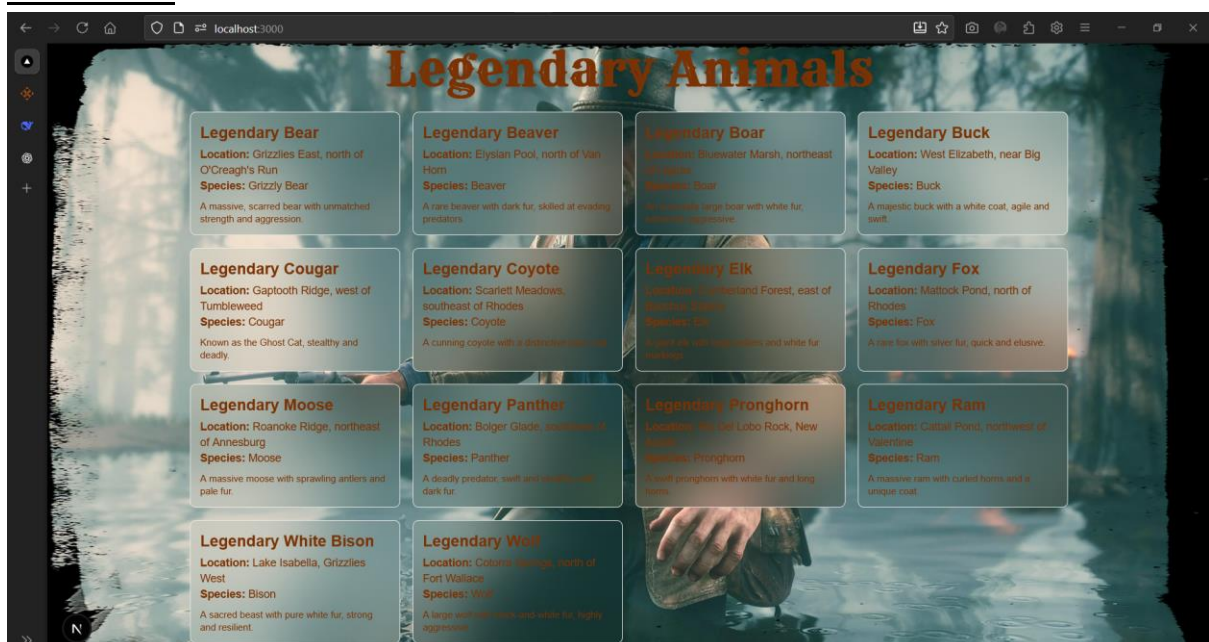
```

```

      desc: "A massive moose with sprawling antlers and pale fur."
    },
    {
      name: "Legendary Panther",
      location: "Bolger Glade, southeast of Rhodes",
      species: "Panther",
      desc: "A deadly predator, swift and stealthy, with dark fur."
    },
    {
      name: "Legendary Pronghorn",
      location: "Rio Del Lobo Rock, New Austin",
      species: "Pronghorn",
      desc: "A swift pronghorn with white fur and long horns."
    },
    {
      name: "Legendary Ram",
      location: "Cattail Pond, northwest of Valentine",
      species: "Ram",
      desc: "A massive ram with curled horns and a unique coat."
    },
    {
      name: "Legendary White Bison",
      location: "Lake Isabella, Grizzlies West",
      species: "Bison",
      desc: "A sacred beast with pure white fur, strong and resilient."
    },
    {
      name: "Legendary Wolf",
      location: "Cotorra Springs, north of Fort Wallace",
      species: "Wolf",
      desc: "A large wolf with black-and-white fur, highly aggressive."
    }
  ];

```

OUTPUT:



Q) Create functional component to render a profile page

CODE (John):

```
"use client";

import Image from "next/image";
import JohnIMG from "../../public/img/Red Dead Redemption John (2).jpg";

export default function John() {
  const profile = {
    name: "John Marston",
    fullName: "John Marston",
    born: "1873",
    died: "1911 (Beecher's Hope, West Elizabeth)",
    family: "Husband to Abigail Roberts, father to Jack Marston",
    role: "Former Van der Linde Gang gunslinger",
    desc: "John Marston was once an outlaw in the Van der Linde gang, known for his gunslinging skills and troubled past. After the gang's collapse, he sought redemption by building a life with Abigail and their son, Jack. Despite his efforts to live peacefully, John was forced into violence once again by the Bureau of Investigation, ultimately leading to his tragic death.",
  };

  return (
    <div className="relative min-h-screen w-full bg-gradient-to-b from-stone-900 via-stone-800 to-stone-900 text-amber-100">
      <div className="relative h-64 w-full">
        <div className="absolute inset-0 flex items-center justify-center">
          <h1 className="text-4xl md:text-6xl font-bold text-amber-400 drop-shadow-lg font-rye">
            John Marston
          </h1>
        </div>
      </div>

      <div className="max-w-4xl mx-auto mt-[-4rem] relative z-10 p-6">
        <div className="bg-stone-800/80 backdrop-blur-lg rounded-2xl shadow-xl p-6 flex flex-col md:flex-row items-center gap-6 border border-stone-700">
          <div className="w-44 h-44 relative flex-shrink-0 rounded-xl overflow-hidden border-2 border-amber-400 shadow-md">
            <Image
              src={JohnIMG}
              alt={profile.name}
              fill
              className="object-cover"
            />
          </div>

          <div className="flex flex-col">
            <h2 className="text-2xl md:text-3xl font-bold text-amber-300">
              {profile.fullName}
            </h2>
            <p className="text-sm italic text-amber-200 mb-2">{profile.role}</p>

            <ul className="space-y-1 text-sm">
              <li>
                <span className="font-semibold">Born:</span> {profile.born}
              </li>
              <li>
                <span className="font-semibold">Died:</span> {profile.died}
              </li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  );
}
```



```

        <li>
          <span className="font-semibold">Family:</span>
{profile.family}
        </li>
      </ul>

      <p className="mt-4 text-sm leading-relaxed text-amber-100">
        {profile.desc}
      </p>
    </div>
  </div>
</div>
</div>
  </div>
);
}

```

CODE (Rendering):

```

import John from "@app/components/John";
export default function Home() {
  return (
    <John />
  );
}

```

OUTPUT:



Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 9

Date Of Performance: 25/09/2025

Aim: Implementation of React useState, useEffect, useContext

Q) Create functional components to demonstrate useState

CODE (Nose):

```
"use client"

import { useState } from "react"

export const Nose = () => {

  const [count, setCount] =
    useState(271000)

  const increment = () => {
    setCount(count + 1)
  }

  const decrement = () => {
    if (count < 271001) {
      alert("You can't go below the
actual number of Jews killed in the
holocaust.")
    }
    else {
      setCount(count - 1)
    }
  }

  return (
    <section className="h-screen flex
flex-col items-center justify-center gap-
4">

      <h2 className="text-2xl font-
bold">How many Jews were killed in the
holocaust?</h2>

      <div className="flex items-
center gap-4">

        <button
          onClick={decrement}
          className="px-4 py-2
bg-red-500 text-2xl text-white rounded-full
hover:bg-red-600"
        >
          -
        </button>

        <span className="text-2xl
font-bold">{count}</span>

        <button
          onClick={increment}
          className="px-4 py-2
bg-green-500 text-2xl text-white rounded-
full hover:bg-green-600"
        >
          +
        </button>

      </div>

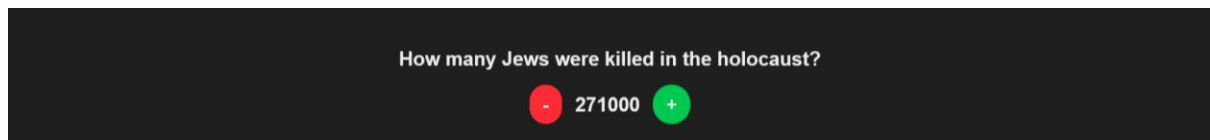
    </section>
  )
}
```

CODE (Rendering):

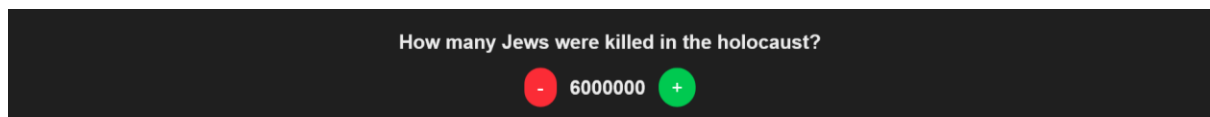
```
import { Nose } from "../components/Nose";
```

```
export default function Home() {
  return (
    <>
      <Nose />
    </>
  );
}
```

OUTPUT:



Initial



Incremented Value (Propaganda)

Q) Create functional components to demonstrate useEffect

CODE (HeilHitler):

```
"use client"

import { useEffect, useState } from "react"

export const HeilHitler = () => {

  const [count, setCount] =
    useState(271000)

  const increment = () => {
    setCount(count + 1)
  }

  const decrement = () => {
    if (count < 271001) {
      alert("You can't go below the
actual number of Jews killed in the
holocaust.")
    }
    else {
      setCount(count - 1)
    }
  }

  useEffect(() => {
    console.log("Heil Hitler Counter:
", count);
  }, [count])

  return (
    <section className="h-screen flex
flex-col items-center justify-center gap-
4">
      <h2 className="text-2xl font-
bold">How many Jews were killed in the
holocaust?</h2>
      <div className="flex items-
center gap-4">
        <button
          onClick={decrement}
          className="px-4 py-2
bg-red-500 text-2xl text-white rounded-full
hover:bg-red-600"
        >
          -
        </button>
        <span className="text-2xl
font-bold">{count}</span>
        <button
          onClick={increment}
          className="px-4 py-2
bg-green-500 text-2xl text-white rounded-
full hover:bg-green-600"
        >
          +
        </button>
      </div>
      <p>Check Console</p>
    </section>
  )
}
```

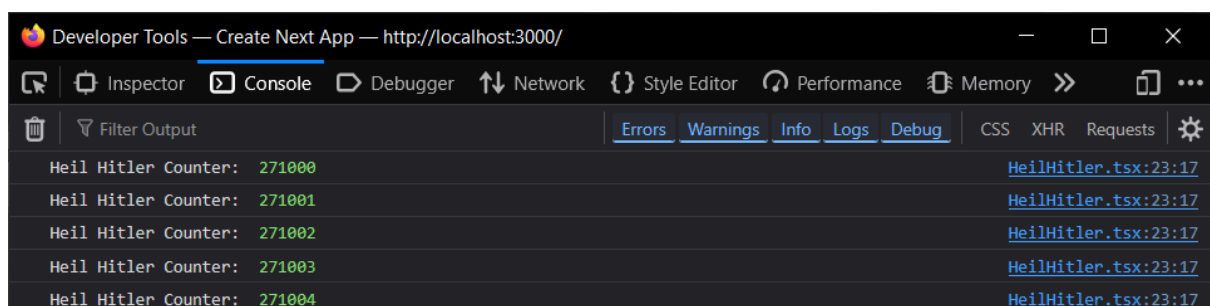
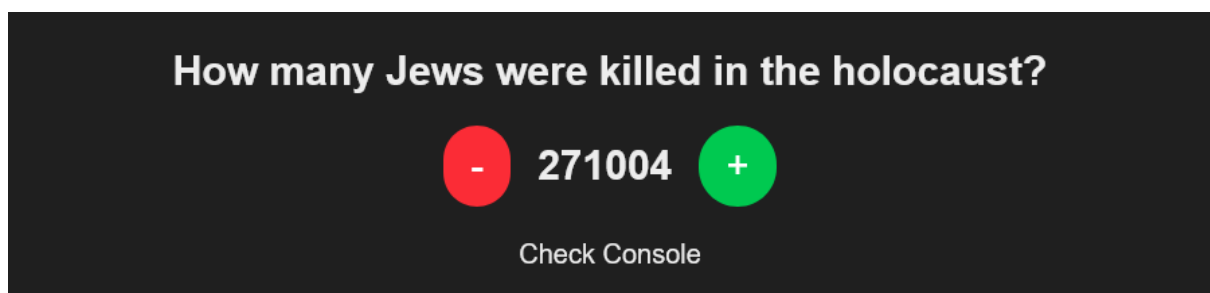
```
}
```

CODE (Rendering):

```
import {HeilHitler} from "../components/HeilHitler";

export default function Home() {
  return (
    <>
      <HeilHitler/>
    </>
  );
}
```

OUTPUT:



Console

Q) Create functional components to demonstrate useContext

CODE (Animals):

"use client"

```
import { useState, createContext,
useContext } from "react"
```

```
const CounterContext = createContext({
  count: 271000,
  increment: () => { },
  decrement: () => { }
})
```

```
const CounterProvider = ({ children }: {
  children: React.ReactNode }) => {
```

```
  const [count, setCount] =
    useState(271000)

  const increment = () => setCount(count
    + 1)
  const decrement = () => {
    if (count <= 271000) {
      alert("You can't go below the
        actual number of Jews killed in the
        holocaust.")
    } else {
      setCount(count - 1)
    }
  }
}
```

```

    }
    return (
      <CounterContext.Provider value={{
count, increment, decrement }}>
        {children}
      </CounterContext.Provider>
    )
  }
  const useCounter = () =>
    useContext(CounterContext)

  export const Counter = () => {
    const { count, increment, decrement } =
    useCounter()
    return (
      <section className="h-screen flex
flex-col items-center justify-center gap-
4">
        <h2 className="text-2xl font-
bold">Gassed Jews Counter</h2>
        <div className="flex items-
center gap-4">

```

```

          <button onClick={decrement}
className="px-4 py-2 bg-red-500 text-white
rounded-full">
            -
          </button>
          <span className="text-2xl
font-bold">{count}</span>
          <button onClick={increment}
className="px-4 py-2 bg-green-500 text-
white rounded-full">
            +
          </button>
        </div>
      </section>
    )
  }
  export const HolocaustCounter = () => (
    <CounterProvider>
      <Counter />
    </CounterProvider>
  )

```

CODE (Rendering):

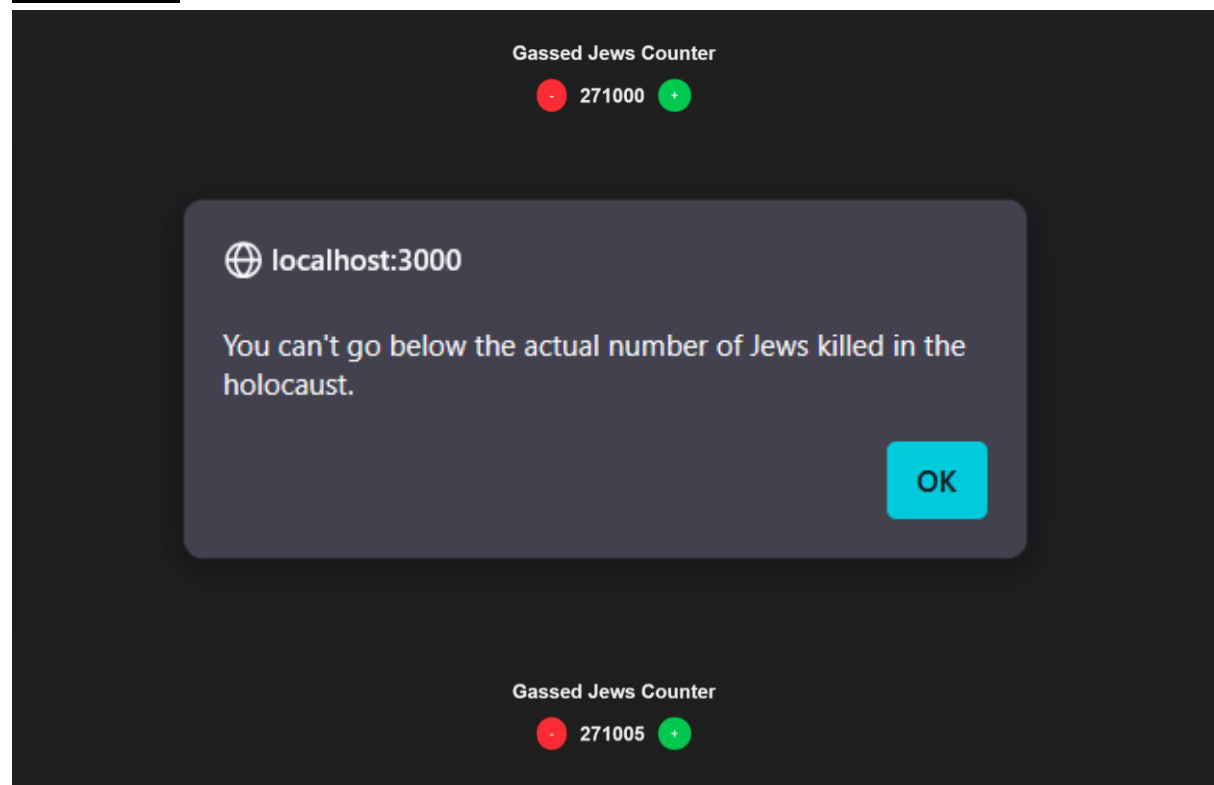
```

import HolocaustCounter from "../components/HolocaustCounter";

export default function Home() {
  return (
    <>
      <HolocaustCounter/>
    </>
  );
}

```

OUTPUT:



Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 10

Date Of Performance: 09/10/2025

Aim: Implementation of React useRef, router, keys etc.

Q) Create functional components to demonstrate router

CODE (Rulers):

```
import { useState, useEffect } from "react";

const HomePage = () => (
  <div className="p-5 text-center">
    <h1 className="text-3xl font-bold mb-4"><img alt="crown icon" data-bbox="268 561 281 574"/> Great Rulers of History</h1>
    <p className="text-lg mb-2">
      Explore the lives and legacies of history's most influential leaders!
    </p>
    <p className="text-lg mb-6">
      From ancient conquerors to military geniuses, discover their stories.
    </p>
    <div className="mt-5">
      <div
        className="grid grid-cols-1 md:grid-cols-3 gap-4 mt-5">
        <div
          className="p-4 bg-blue-50 rounded-lg">
          <h4
            className="font-semibold text-lg">Military Genius</h4>
```

```
          <p>Strategic brilliance and battlefield tactics</p>
        </div>
        <div
          className="p-4 bg-green-50 rounded-lg">
          <h4
            className="font-semibold text-lg">Empire Building</h4>
          <p>Conquests and territorial expansion</p>
        </div>
        <div
          className="p-4 bg-purple-50 rounded-lg">
          <h4
            className="font-semibold text-lg">Lasting Legacy</h4>
          <p>Impact on civilization and culture</p>
        </div>
      </div>
    </div>
  );

const NapoleonPage = () => (
  <div className="p-5 text-center">
```

```
    <h1 className="text-3xl font-bold mb-4"><img alt="crown icon" data-bbox="814 486 827 499"/> Napoleon Bonaparte</h1>
    <p className="text-lg mb-6">
      The French military genius who dominated Europe
    </p>
    <div className="max-w-3xl mx-auto p-5 bg-orange-50 rounded-lg text-left">
      <h3
        className="text-xl font-semibold mb-3">
        Historical Significance
      </h3>
      <p className="mb-4">
        Napoleon Bonaparte (1769-1821) was a French military commander and political leader who rose to prominence during the French Revolution and led several successful campaigns during the Revolutionary Wars.
      </p>
```

```

        <h3
className="text-xl font-
semibold mb-3">Key
Achievements</h3>
        <ul
className="list-disc list-
inside space-y-2 mb-4">
            <li>👑
Crowned Emperor of the French
in 1804</li>
            <li>
                🏆 Won
numerous battles including
Austerlitz and Marengo
            </li>
            <li>📜
Established the Napoleonic
Code</li>
            <li>🌍
Controlled most of
continental Europe</li>
        </ul>

        <div
className="mt-4 p-3 bg-blue-
100 rounded-lg">
            <strong>Famous
Quote:</strong> "Impossible
is a word to be found
            only in the
dictionary of fools."
        </div>
    </div>
);

const CaesarPage = () => (
    <div className="p-5 text-
center">
        <h1 className="text-
3xl font-bold mb-4">🏰 Julius
Caesar</h1>
        <p className="text-lg
mb-6">
            The legendary
Roman general and statesman
        </p>

        <div className="grid
grid-cols-1 md:grid-cols-3
gap-5 mt-5">
            <div
className="p-5 border-2
border-brown-600 rounded-lg
bg-gray-50">
                <h3
className="text-lg font-
semibold mb-2">👑
Military Campaigns
                </h3>
                <p
className="mb-3">

```

```

Conqueror
of Gaul and victor of the
Civil War
        </p>
        <div
className="text-xl font-bold
text-brown-600">58-50
BC</div>
        <div
className="mt-3">
            <strong>6
allied Wars:</strong> 8 years
of conquest
        </div>
        <div
className="p-5 border-2
border-red-700 rounded-lg bg-
gray-50">
            <h3
className="text-lg font-
semibold mb-2">🏰
Political Reforms
        </h3>
        <p
className="mb-3">Transformed
the Roman Republic</p>
        <div
className="text-xl font-bold
text-red-700">49-44 BC</div>
        <div
className="mt-3">
            <strong>D
ictator:</strong> Appointed
for life
        </div>
        <div
className="p-5 border-2
border-gray-700 rounded-lg
bg-gray-50">
            <h3
className="text-lg font-
semibold mb-2">🏰 Legacy</h3>
            <p
className="mb-3">Julian
Calendar and literary
works</p>
            <div
className="text-xl font-bold
text-gray-700">
                Eternal
Impact
            </div>
            <div
className="mt-3">
                <strong>A
ssassinated:</strong> Ides of
March, 44 BC
            </div>
        </div>
    </div>

```

```

        <div className="mt-5
p-4 bg-blue-50 rounded-lg">
            <strong>Famous
Quote:</strong> "Veni, Vidi,
Vici" (I came, I saw, I
conquered)
        </div>
    </div>
);

const RouterDemo = () => {
    const [currentPath,
setCurrentPath] =
useState(window.location.path
name);

    useEffect(() => {
        const handlePopState
= () => {
            setCurrentPath(wi
ndow.location.pathname);
        };

        window.addEventListen
er("popstate",
handlePopState);
        return () =>
window.removeEventListener("p
opstate", handlePopState);
    }, []);

    const navigate = (path)
=> {
        window.history.pushSt
ate({}, "", path);
        setCurrentPath(path);
    };

    const renderComponent =
() => {
        switch (currentPath)
{
            case "/":
                return
<HomePage />;
            case "/napoleon":
                return
<NapoleonPage />;
            case "/caesar":
                return
<CaesarPage />;
            case "/hannibal":
                return
<HannibalPage />;
            default:
                return (
                    <div
className="p-10 text-center">
                        <h1
className="text-3xl font-bold
mb-4">
                            4
                            04 - Ruler Not Found
                        </h1>

```

```

        className="text-lg mb-6">
            T
            he historical ruler you're
            looking for doesn't
            e
            xist in our records.
        </p>
        <butt
            on
            o
            nClick={() => navigate("/")}>
            c
            lassName="px-5 py-3 bg-brown-
            600 text-white border-none
            rounded-lg hover:bg-brown-700
            transition-colors"
        >
            R
            eturn to Hall of Rulers
        </but
            ton>
        </div>
    );
    };

    const isActive = (path)
    => currentPath === path;

    return (
        <div className="p-5
            m-3 rounded-lg bg-gray-50">
            <nav
                className="mb-5 p-4 bg-amber-
                50 rounded-lg border border-
                amber-200">
                <h3
                    className="text-xl font-
                    semibold mb-3">
                    Gallery
                of Great Leaders
            </h3>
            <div
                className="flex gap-3 flex-
                wrap">
                <button
                    onClick={() => navigate("/")}>
                    class
                    Name={`px-4 py-3 text-white
                    border-none rounded-lg
                    cursor-pointer transition-
                    colors ${
                        isActive("/")
                            ? "bg-brown-600"
                            : "bg-amber-600 hover:bg-
                            amber-700"
                    }`}
                    >
                    <img alt="Home icon" data-bbox="565 288 585 303"/>
                    Home
                </button>
                <button
                    onClick={() =>
                        navigate("/napoleon")
                    }>
                    class
                    Name={`px-4 py-3 text-white
                    border-none rounded-lg
                    cursor-pointer transition-
                    colors ${
                        isActive("/napoleon")
                            ? "bg-brown-600"
                            : "bg-amber-600 hover:bg-
                            amber-700"
                    }`}
                    >
                    <img alt="Napoleon icon" data-bbox="565 555 585 570"/>
                    Napoleon
                </button>
                <button
                    onClick={() =>
                        navigate("/caesar")
                    }>
                    class
                    Name={`px-4 py-3 text-white
                    border-none rounded-lg
                    cursor-pointer transition-
                    colors ${
                        isActive("/caesar")
                            ? "bg-brown-600"
                            : "bg-amber-600 hover:bg-
                            amber-700"
                    }`}
                    >
                    <img alt="Caesar icon" data-bbox="838 225 858 240"/>
                    Caesar
                </button>
            </div>
            </nav>
            <div
                className="mb-4 p-3 bg-amber-
                50 rounded-lg border border-
                amber-200">
                <strong>Curre
                nt Historical
                Figure:</strong>{" "}
                <code
                    className="bg-white px-2 py-1
                    rounded">
                    {currentP
                    ath}
                </code>
            </div>
            <div
                className="border border-
                amber-200 rounded-lg min-h-80
                bg-white">
                {renderCompon
                ent()}
            </div>
        </div>
    );
    };

    export default RouterDemo;

```

CODE (Rendering):

```

import Rulers from "../components/Rulers";

function App() {
    return (
        <Rulers />
    );
}

export default App;

```


OUTPUT:

Gallery of Great Leaders

[Home](#) [Napoleon](#) [Caesar](#) [Hannibal](#)

Current Historical Figure: /caesar

Julius Caesar

The legendary Roman general and statesman

Military Campaigns

Conqueror of Gaul and victor of the Civil War

58-50 BC

Gaulic Wars: 8 years of conquest

Political Reforms

Transformed the Roman Republic

49-44 BC

Dictator: Appointed for life

Legacy

Julian Calendar and literary works

Eternal Impact

Assassinated: Ides of March, 44 BC

Famous Quote: "Veni, Vidi, Vici" (I came, I saw, I conquered)

Gallery of Great Leaders

[Home](#) [Napoleon](#) [Caesar](#) [Hannibal](#)

Current Historical Figure: /napoleon





Napoleon Bonaparte

The French military genius who dominated Europe

Historical Significance

Napoleon Bonaparte (1769-1821) was a French military commander and political leader who rose to prominence during the French Revolution and led several successful campaigns during the Revolutionary Wars.

Key Achievements

-  Crowned Emperor of the French in 1804
-  Won numerous battles including Austerlitz and Marengo
-  Established the Napoleonic Code
-  Controlled most of continental Europe

Famous Quote: "Impossible is a word to be found only in the dictionary of fools."

Q) Create functional components to demonstrate useRef

CODE (FocusJasGews):

```
import { useRef } from "react";

function FocusJasGews() {
  const inputRef = useRef(null);

  const handleClick = () => {
    inputRef.current.focus();
  };

  return (
    <div>
```

```
      <input ref={inputRef}
        type="text" placeholder="Jas Gews" />
      <button
        onClick={handleClick}>Click to Jas
        Gews</button>
    </div>
  );
}

export default FocusJasGews;
```

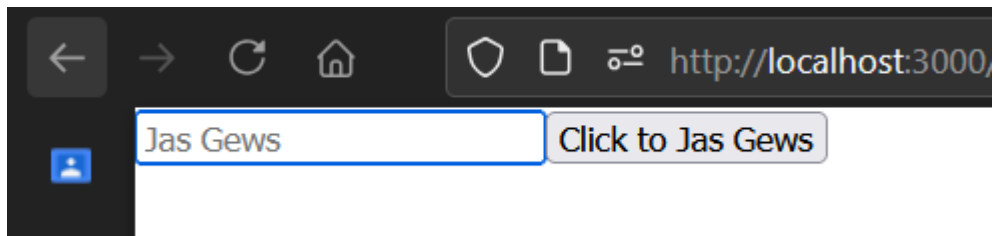
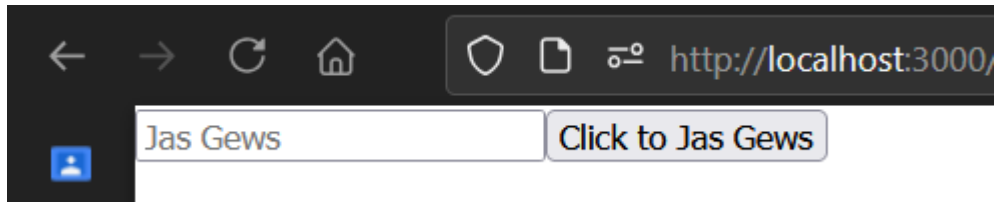
CODE (Rendering):

```
import FocusJasGews from "../components/FocusJasGews";
```

```
function App() {
  return (
    <FocusJasGews />
  );
}

export default App;
```

OUTPUT:



Q) Create functional components to demonstrate keys

CODE (JewTattoo):

```
import React, { useState }
from "react";

const JewTatto = () => {
  const [items,
setItems] = useState([
    { id: 1, name:
"Goldberg", number: 496242
},
    { id: 2, name:
"Rosenstein", number:
177854 },
    { id: 3, name:
"Silverman", number:
823659 },
    { id: 4, name:
"Feinberg", number: 896523
},
    { id: 5, name:
"Abramowitz", number:
564175 },
  ]);

  return (
    <div style={{
padding: "20px" }}>
      <h2>Jew Tatto
Number Demonstration</h2>
      <div>
        <h3>List
with Proper Keys:</h3>
        <ul>
          {items
.map((item) => (
            <li
              key={item.id}
              style={{
                color: item.color,
                padding: "5px",
                margin: "5px 0",
              }}
            >
              {item.name} (Number:
{item.number})
            </li>
          ))}
        </ul>
      </div>
    </div>
  );
};

export default JewTatto;
```

CODE (Rendering):

```
import JewTatto from "../components/JewTatto";

function App() {
  return (
    <JewTatto />
  );
}

export default App;
```

OUTPUT:

Jew Tattoo Number Demonstration

List with Proper Keys:

- Goldberg (Number: 496242)
- Rosenstein (Number: 177854)
- Silverman (Number: 823659)
- Feinberg (Number: 896523)
- Abramowitz (Number: 564175)

Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 11

Date Of Performance: 09/10/2025

Aim: Implementation of Nodejs Server side Async Call back Programming

CODE (package.json):

```
{
  "name": "nigga-chan",
  "version": "1.0.0",
  "description":
    "Node.js Async Callback
    Demonstration",
  "main": "main.js",
  "scripts": {
    "start": "node
    main.js",
    "test": "echo
    \\\"Run: node main.js\\\" &&
    exit 1"
  },
  "keywords": [
    "nodejs",
    "async",
    "callbacks",
    "demonstration"
  ],
  "author":
    "Developer",
  "license": "MIT"
}
```

CODE (main.js):

```
const fs = require('fs');
const http = require('http');

function readUserData(userId,
callback) {
  console.log(`Reading data
  for user ${userId}...`);
  fs.readFile(`user-
  ${userId}.txt`, 'utf8', (err,
  data) => {
    if (err) {
      return
      callback(err);
    }
    try {
      const userData =
      JSON.parse(data);
      callback(null,
      userData);
    } catch (parseErr) {
      callback(parseErr
      );
    }
  });
}

function
writeUserData(userId,
userData, callback) {
  console.log(`Writing data
  for user ${userId}...`);
  fs.writeFile(`user-
  ${userId}.txt`,
  JSON.stringify(userData,
  null, 2), (err) => {
    if (err) {
      return
      callback(err);
    }
  });
}

function
fetchDataFromAPI(callback) {
  console.log('Making API
  request...');
  const options = {
    hostname:
      'jsonplaceholder.typicode.com',
    path: '/users/1',
    method: 'GET',
    headers: {
      'Content-Type':
      'application/json'
    }
  };
  http.get(options, (res) => {
    let data = '';
    res.on('data', (chunk) => {
      data += chunk;
    });
    res.on('end', () => {
      callback(null, `User
      ${userId} data saved
      successfully`);
    });
  });
}
```

```

    };

    const req =
    http.request(options, (res)
    => {
        let data = '';

        res.on('data',
        (chunk) => {
            data += chunk;
        });

        res.on('end', () => {
            try {
                const
                parsedData =
                JSON.parse(data);
                callback(null
                , parsedData);
            } catch
            (parseErr) {
                callback(pars
                eErr);
            }
        });
    });

    req.on('error', (err) =>
    {
        callback(err);
    });

    req.setTimeout(5000, ()
    => {
        req.destroy();
        callback(new
        Error('Request timeout'));
    });

    req.end();
}

function
demonstrateCallbacks() {
    console.log('=== Starting
    Async Callback Demonstration
    ===\n');

    readUserData(1, (err,
    userData) => {
        if (err) {
            console.error('
            Read Error:', err.message);
        } else {
            console.log('
            Read Success:', userData);
        }

        const newUser = {
            id: 3,
            name: 'Charlie',
            age: 28,
            email:
            'charlie@example.com',
            timestamp: new
            Date().toISOString()

```

```

    });

    writeUserData(3,
    newUser, (err, result) => {
        if (err) {
            console.error
            (' Write Error:',
            err.message);
        } else {
            console.log('
            Write Success:', result);

            readUserData(
            3, (err, verifiedData) => {
                if (err)
                {
                    conso
                    le.error(' Verify Error:',
                    err.message);
                } else {
                    conso
                    le.log(' Verify Success:',
                    verifiedData);
                }
            });
        }
    });

    setTimeout(() => {
        console.log('\n---
        Testing HTTP API Call ---');
        fetchDataFromAPI((err
        , apiData) => {
            if (err) {
                console.error
                (' API Error:', err.message);
            } else {
                console.log('
                API Success:');
                console.log('
                User:', apiData.name);
                console.log('
                Email:', apiData.email);
                console.log('
                Company:',
                apiData.company?.name);
            }
        }, 1000);

        setTimeout(() => {
            console.log('\n---
            Testing Error Handling ---');
            readUserData(999,
            (err, data) => {
                if (err) {
                    console.log('
                    Proper Error Handling - File
                    not found:', err.message);
                }
            }, 2000);
        }
    }

```

```

const server =
http.createServer((req, res)
=> {
    if (req.url === '/user/1'
    && req.method === 'GET') {
        readUserData(1, (err,
        userData) => {
            if (err) {
                res.writeHead
                (404, { 'Content-Type':
                'application/json' });
                res.end(JSON.
                stringify({ error: 'User not
                found' }));
            } else {
                res.writeHead
                (200, { 'Content-Type':
                'application/json' });
                res.end(JSON.
                stringify(userData));
            }
        });
    }
    else if (req.url ===
    '/api-data' && req.method ===
    'GET') {
        fetchDataFromAPI((err
        , apiData) => {
            if (err) {
                res.writeHead
                (500, { 'Content-Type':
                'application/json' });
                res.end(JSON.
                stringify({ error: 'API call
                failed' }));
            } else {
                res.writeHead
                (200, { 'Content-Type':
                'application/json' });
                res.end(JSON.
                stringify(apiData));
            }
        });
    }
    else {
        res.writeHead(200, {
        'Content-Type': 'text/html'
        });
        res.end(`
        <html>
            <body>
                <h1>Node.
                js Async Callback Demo</h1>
                <p>Endpoi
                nts:</p>
                <ul>
                    <li><
                    a href="/user/1">/user/1</a>
                    - Get User 1 Data</li>
                    <li><
                    a href="/api-data">/api-
                    data</a> - Get API Data</li>
                </ul>
            </body>
        </html>
        `);
    }
});

```

```

    }
  });

  const PORT = 3000;
  server.listen(PORT, () => {

    console.log(`Server
    running at
    http://localhost:${PORT}`);
    demonstrateCallbacks();
  });

  module.exports = {
    readUserData,
    writeUserData,
    fetchDataFromAPI
  };

```

OUTPUT:

```


Abdurrahman Qureshi@DESKTOP-H2RV5MQ MINGW64 /d/Degree/SEM 5/INP/Experiments/EXP11 (master)
$ node main.js
Server running at http://localhost:3000
=== Starting Async Callback Demonstration ===

Reading data for user 1...
Read Success: { id: 1, name: 'Alice', age: 30, email: 'alice@example.com' }
Writing data for user 3...
Write Success: User 3 data saved successfully
Reading data for user 3...
Verify Success: {
  id: 3,
  name: 'Charlie',
  age: 28,
  email: 'charlie@example.com',
  timestamp: '2025-10-09T12:37:24.626Z'
}

--- Testing HTTP API Call ---
Making API request...
API Success:
  User: Leanne Graham
  Email: Sincere@april.biz
  Company: Romaguera-Crona

--- Testing Error Handling ---
Reading data for user 999...
Proper Error Handling - File not found: ENOENT: no such file or directory, open 'D:\Degree\SEM 5\INP\Experiments\EXP11\user-999.txt'

```



← → ↻ 🏠 🛡️ 📄 🔍 http://localhost:3000/user/3

Node.js Async Callback Demo by Nigga-Chan

Endpoints:

- </user/1> - Get User 1 Data
- </api-data> - Get API Data

← → ↻ 🏠 http://localhost:3000/api-data

JSON Raw Data Headers

Save Copy Collapse All Expand All 🔍 Filter JSON

```
{
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
  "email": "Sincere@april.biz",
  "address": {
    "street": "Kulas Light",
    "suite": "Apt. 556",
    "city": "Gwenborough",
    "zipcode": "92998-3874",
    "geo": {
      "lat": "-37.3159",
      "lng": "81.1496"
    }
  },
  "phone": "1-770-736-8031 x56442",
  "website": "hildegard.org",
  "company": {
    "name": "Romaguera-Crona",
    "catchPhrase": "Multi-layered client-server neural-net",
    "bs": "harness real-time e-markets"
  }
}
```

← → ↻ 🏠 http://localhost:3000/user/1

JSON Raw Data Headers

Save Copy Collapse All Expand All 🔍 Filter JSON

```
{
  "id": 1,
  "name": "Alice",
  "age": 30,
  "email": "alice@example.com"
}
```

Name: Abdur R. Qureshi

Roll No: 242466

Practical No: 12

Date Of Performance: 09/10/2025

Aim: Implementation of Nodejs Express Applications.

Q) Create functional components to demonstrate

CODE (app.js):

```
const express = require('express');
const path = require('path');
const app = express();
const PORT = process.env.PORT || 3000;

app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

app.use(express.static(path.join(__dirname, 'public')));

const napoleonData = {
  title: "Napoleon Bonaparte - Emperor of the French",
  sections: {
    earlyLife: {
      title: "Early Life",
      content: "Napoleon Bonaparte was born on August 15, 1769, in Ajaccio, Corsica. He attended military school in France and quickly rose through the ranks of the military during the French Revolution."
    },
    militaryCampaigns: {
      title: "Military Campaigns",
      content: "Napoleon led successful campaigns in the Italian Peninsula and Egypt, establishing his reputation as a brilliant military strategist. His most famous victories include Austerlitz, Jena, and Friedland."
    },
    emperor: {
      title: "Emperor of the French",
      content: "In 1804, Napoleon crowned himself Emperor of the French. He implemented
```

```
the Napoleonic Code, which influenced legal systems worldwide, and reformed France's education system."
    },
    exile: {
      title: "Exile and Death",
      content: "After defeat in the War of the Sixth Coalition, Napoleon was exiled to Elba in 1814. He escaped in 1815 but was defeated at Waterloo. He died in exile on Saint Helena in 1821."
    }
  },
  timeline: [
    { year: "1769", event: "Born in Corsica" },
    { year: "1796", event: "Marries Joséphine de Beauharnais" },
    { year: "1799", event: "Becomes First Consul of France" },
    { year: "1804", event: "Crowned Emperor" },
    { year: "1812", event: "Invades Russia" },
    { year: "1814", event: "Exiled to Elba" },
    { year: "1815", event: "Defeated at Waterloo" },
    { year: "1821", event: "Dies in Saint Helena" }
  ]
};

app.get('/', (req, res) => {
  res.render('index', { data: napoleonData });
});
```



```
});

app.listen(PORT, () => {
```

```
console.log(`Napoleon SPA running on
http://localhost:${PORT}`);
});
```

CODE (views/index.js)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>
    <%= data.title %>
  </title>
  <link rel="preconnect"
href="https://fonts.googleapis.com">
  <link rel="preconnect"
href="https://fonts.gstatic.com" crossorigin>
  <link
    href="https://fonts.googleapis.com/css2?
family=Playfair+Display:wght@400;500;600;700&fam
ily=Inter:wght@300;400;500;600&display=swap"
    rel="stylesheet">
  <link rel="stylesheet"
href="/css/style.css">
</head>
<body>
  <header>
    <nav class="navbar">
      <div class="container">
        <div class="nav-brand">
          <h1>
            <%= data.title %>
          </h1>
        </div>
        <ul class="nav-links">
          <li><a href="#early-
life">Early Life</a></li>
          <li><a
href="#military">Military</a></li>
          <li><a
href="#emperor">Emperor</a></li>
          <li><a
href="#exile">Exile</a></li>
          <li><a
href="#timeline">Timeline</a></li>
          <li><a
href="#gallery">Gallery</a></li>
        </ul>
      </div>
    </nav>
  </header>
  <main>
    <section class="hero">
      <div class="hero-content">
        <h2>The Rise and Fall of an
Emperor</h2>
        <p>Explore the life and legacy
of one of history's most fascinating military
leaders and rulers</p>
        <div class="hero-stats">
          <div class="stat">
```

```

          <span class="stat-
number">52</span>
          <span class="stat-
label">Battles Fought</span>
        </div>
        <div class="stat">
          <span class="stat-
number">15</span>
          <span class="stat-
label">Years as Emperor</span>
        </div>
        <div class="stat">
          <span class="stat-
number">12M</span>
          <span class="stat-
label">Square Km Empire</span>
        </div>
      </div>
      <div class="hero-scroll">
        <span>↓</span>
      </div>
    </section>
    <section id="early-life" class="content-
section with-image">
      <div class="container">
        <div class="section-header">
          <h2>
            <%=
data.sections.earlyLife.title %>
          </h2>
          <div class="section-
divider"></div>
        </div>
        <div class="content-grid">
          <div class="content-text">
            <p>
              <%=
data.sections.earlyLife.content %>
            </p>
            <div class="key-facts">
              <h3>Key Facts</h3>
              <ul>
                <li>Born in
Ajaccio, Corsica on August 15, 1769</li>
                <li>Attended
military school in Brienne, France</li>
                <li>Graduated
from École Militaire in Paris at age 16</li>
                <li>Spoke French
with a Corsican accent throughout his life</li>
              </ul>
            </div>
          </div>
          <div class="content-image">
            
        <p class="image-
caption">Napoleon's birthplace in Ajaccio,
Corsica</p>
        </div>
    </div>
</section>
<section id="military" class="content-
section with-image">
    <div class="container">
        <div class="section-header">
            <h2>
                <%=
data.sections.militaryCampaigns.title %>
            </h2>
            <div class="section-
divider"></div>
        </div>
        <div class="content-grid
reverse">
            <div class="content-text">
                <p>
                    <%=
data.sections.militaryCampaigns.content %>
                </p>
                <div class="battles-
list">
                    <h3>Major
Battles</h3>
                    <div class="battle-
item">
                        <span
class="battle-name">Austerlitz</span>
                        </div>
                        <div class="content-image">
                            
                            <p class="image-
caption">Napoleon Bonaparte in his military
uniform</p>
                        </div>
                    </div>
                </div>
            </section>
            <section id="emperor" class="content-
section with-image">
                <div class="container">
                    <div class="section-header">
                        <h2>
                            <%=
data.sections.emperor.title %>
                        </h2>
                        <div class="section-
divider"></div>
                    </div>
                    <div class="content-grid">
                        <div class="content-text">
                            <p>

```

```

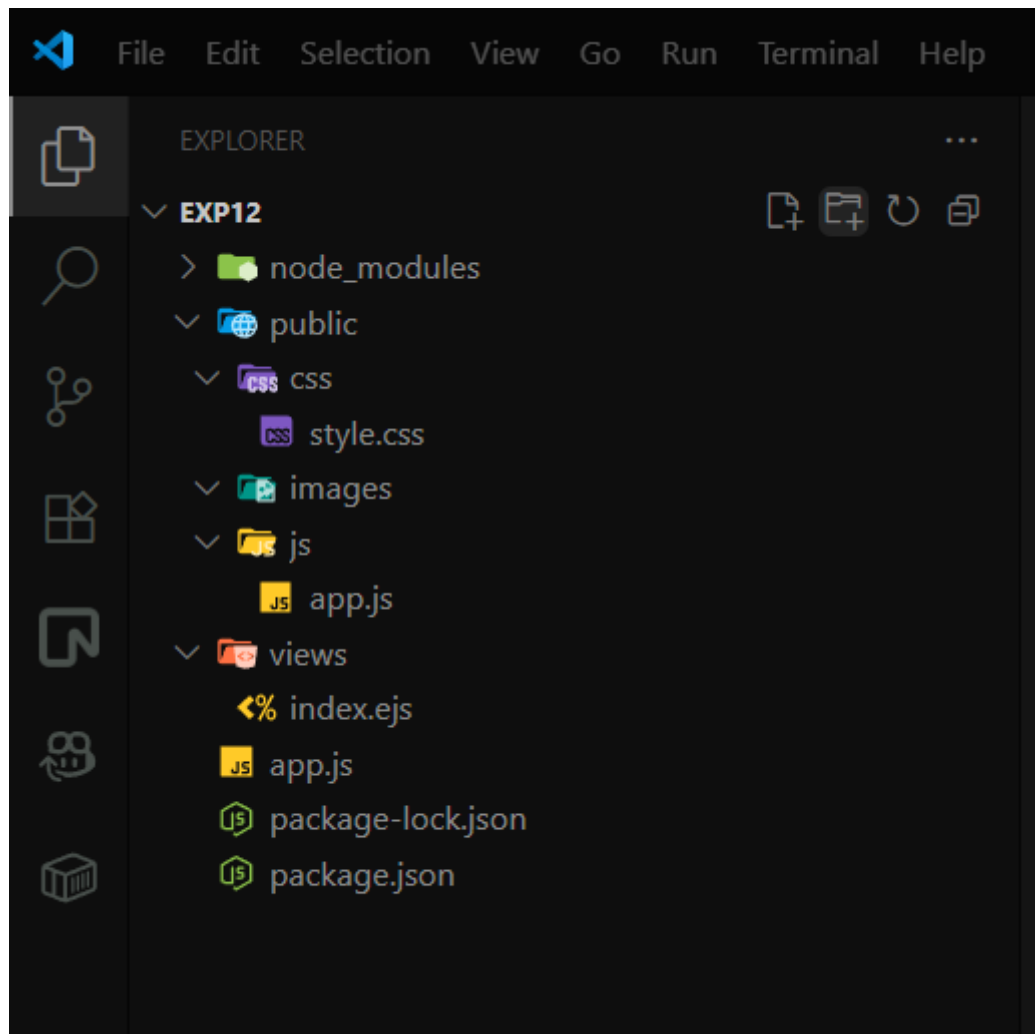
                <%=
data.sections.emperor.content %>
            </p>
        </div>
    </div>
    class="achievements">
        <h3>Key
Achievements</h3>
        <div
class="achievement-grid">
            <div
class="achievement-item">
                <div
class="achievement-icon"><img alt="Key icon" /></div>
                <h4>Napoleon
ic Code</h4>
                <p>Legal
system that influenced civil law worldwide</p>
            </div>
            <div
class="achievement-item">
                <div
class="achievement-icon"><img alt="Bank icon" /></div>
                <h4>Bank of
France</h4>
                <p>Establish
ed central banking system</p>
            </div>
            <div
class="achievement-item">
                <div
class="achievement-icon"><img alt="Education icon" /></div>
                <h4>Educatio
n Reform</h4>
                <p>Created
lycée system for secondary education</p>
            </div>
        </div>
        <div class="content-image">
            
            <p class="image-
caption">Napoleon's coronation as Emperor of the
French</p>
        </div>
    </div>
</div>
</section>
<section id="exile" class="content-
section with-image">
    <div class="container">
        <div class="section-header">
            </section>
            <section id="gallery" class="content-
section">
                <div class="container">
                    <div class="section-header">
                        <h2>Historical Gallery</h2>
                    <div class="section-
divider"></div>

```

```
</div>
<div class="gallery-grid">
  <div class="gallery-item">
    
    <div class="gallery-
caption">Napoleon Crossing the Alps</div>
  </div>
  <div class="gallery-item">
    
    <div class="gallery-
caption">Battle of Austerlitz</div>
  </div>
  <div class="gallery-item">
    
    <div class="gallery-
caption">Napoleon at Waterloo</div>
  </div>
  <div class="gallery-item">
    
    <div class="gallery-
caption">Emperor Napoleon I</div>
  </div>
</div>
</section>
<section id="legacy" class="content-
section bg-dark">
  <div class="container">
    <div class="section-header">
      <h2>Napoleon's Legacy</h2>
    </div>
    <div class="section-
divider"></div>
  </div>
  <div class="legacy-content">
    <div class="quote">
      <blockquote>
        "Impossible is a
word to be found only in the dictionary of
fools."
      </blockquote>
      <cite>- Napoleon
Bonaparte</cite>
    </div>
  </div>
</section>
```

```
</blockquote>
</div>
<div class="legacy-facts">
  <h3>Lasting Impact</h3>
  <ul>
    <li>Napoleonic Code
influenced legal systems worldwide</li>
    <li>Reformed
education and administrative systems in
France</li>
    <li>Spread
revolutionary ideals across Europe</li>
    <li>Inspired
military tactics and strategy for
generations</li>
  </ul>
</div>
</div>
</section>
</main>
<footer>
  <div class="container">
    <div class="footer-content">
      <div class="footer-section">
        <h3>Napoleon Bonaparte</h3>
        <p>Emperor of the French
(1804-1814/1815)</p>
      </div>
      <div class="footer-section">
        <h4>Explore More</h4>
        <ul>
          <li><a href="#early-
life">Early Life</a></li>
          <li><a
href="#military">Military Career</a></li>
          <li><a
href="#emperor">As Emperor</a></li>
          <li><a
href="#timeline">Timeline</a></li>
        </ul>
      </div>
      <div class="footer-section">
        <h4>Resources</h4>
        <ul>
          <li><a
href="#">Historical Documents</a></li>
          <li><a href="#">Museum
Collections</a></li>
          <li><a href="#">Further
Reading</a></li>
        </ul>
      </div>
    </div>
  </div>
  <div class="footer-bottom">
    <p>&copy; 2024 Napoleon
Bonaparte Historical Archive. Educational
purposes only.</p>
    <p>All images from Wikimedia
Commons - Public Domain</p>
  </div>
</div>
</script>
</body>
</html>
```

Folder Structure:



OUTPUT:

