

## Exp 1:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple HTML Demo Page</title>
    <meta charset="UTF-8" />
  </head>
  <body>
    <h1 align="center">My Static Website Demo</h1>

    <p>
      This page demonstrates basic <b>formatting</b> elements,
      <i>hyperlinks</i>, images, and more.
    </p>
    <hr />

    <h2>1. Hyperlink</h2>
    <p>This is a <a href="https://www.google.com">hyperlink to Google</a>.</p>
    <hr />

    <h2>2. Image</h2>
    <p>This is a simple image:</p>
    
    <hr />

    <h2>3. List</h2>
    <p>This is an unordered list:</p>
    <ul>
      <li>List item 1</li>
      <li>List item 2</li>
      <li>List item 3</li>
    </ul>
    <hr />

    <h2>4. Table</h2>
    <p>This is a simple table:</p>
    <table border="1">
      <tr>
        <th>First Name</th>
        <th>Last Name</th>
      </tr>
      <tr>
        <td>John</td>
        <td>Doe</td>
      </tr>
      <tr>
        <td>Jane</td>
        <td>Smith</td>
      </tr>
```

</table>

<hr />

## <h2>5. Form</h2>

<p>This is a simple form:</p>

<form>

<label for="fullName">Full Name:</label>

<input type="text" id="fullName" name="fullName" />

<br /><br />

<p>Primary Language:</p>

<input type="radio" id="python" name="primaryLanguage" value="Python" />

<label for="python">Python</label><br />

<input type="radio" id="java" name="primaryLanguage" value="Java" />

<label for="java">Java</label><br /><br />

<p>Additional Subjects:</p>

<input type="checkbox" id="math" name="subjects" value="math" />

<label for="math">Math</label><br />

<input type="checkbox" id="science" name="subjects" value="science" />

<label for="science">Science</label><br /><br />

<label for="comments">Comments:</label><br />

<textarea id="comments" name="comments" rows="4" cols="40"></textarea>

<br /><br />

<input type="submit" value="Register" />

</form>

<hr />

## <h2>6. Frames (iframe)</h2>

<p>This iframe embeds another website (example.com):</p>

<iframe src="https://example.com" width="400" height="200">

Your browser does not support iframes.

</iframe>

<hr />

## <h2>7. Multimedia (Video)</h2>

<p>This is an HTML5 video player:</p>

<video width="320" height="240" controls>

<source

src="https://www.w3schools.com/html/mov\_bbb.mp4"

type="video/mp4"

/>

Your browser does not support the video tag.

</video>

</body>

</html>

## Exp 2:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>CSS3 Demo Page</title>

  <style>
    body {
      background-color: #f9f9f9;
      color: #333;
    }

    h1,
    h2 {
      color: #6f4e37;
    }

    .highlight {
      background-color: #fff8e1;
    }

    #footer-note {
      text-align: center;
      font-size: 0.9em;
      margin-top: 40px;
    }

    a:hover {
      color: #d9534f;
    }

    ul {
      background-color: #ffffff;
      border: 1px solid #ddd;
      padding-top: 10px;
      padding-bottom: 10px;
    }

    table {
      width: 60%;
      border-collapse: collapse;
      margin-top: 20px;
    }

    th,
    td {
      border: 1px solid #aaa;
      padding: 8px;
    }

    th {
```

```

    background-color: #6f4e37;
    color: white;
}
</style>
</head>
<body>
<h1>CSS3 Demonstration Page</h1>
<p>
    This page demonstrates the core CSS concepts you requested.
    <a href="#">This is a link</a> that changes color on hover.
</p>
<p class="highlight">
    This paragraph has a "highlight" class, so it gets a special
    <b>background-color</b>.
</p>

<hr />

<h2>Styled List</h2>
<p>Here is a <b>list</b> styled with CSS:</p>
<ul>
    <li>Espresso</li>
    <li>Cappuccino</li>
    <li>Iced Latte</li>
</ul>

<hr />

<h2>Styled Table</h2>
<p>Here is a <b>table</b> styled with CSS:</p>
<table>
    <tr>
        <th>Coffee</th>
        <th>Small Price</th>
        <th>Large Price</th>
    </tr>
    <tr>
        <td>Classic Espresso</td>
        <td>₹180</td>
        <td>₹220</td>
    </tr>
    <tr>
        <td>Frothy Cappuccino</td>
        <td>₹250</td>
        <td>₹300</td>
    </tr>
</table>

<hr />

<p id="footer-note">
    This paragraph is styled using an ID <b>selector</b>.

```

</p>  
</body>  
</html>

### Exp 3:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Bootstrap Components Demo</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
      rel="stylesheet"
    />
  </head>
  <body class="bg-light">
    <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">DemoSite</a>
        <button
          class="navbar-toggler"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#navbarNav"
        >
          <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" href="#">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#">About</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#">Contact</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>

    <div class="container mt-5 p-4 bg-white rounded shadow-sm">
      <h1 class="mb-4">Bootstrap Demo Page</h1>
      <p>
        This page demonstrates a simple Bootstrap Form, Buttons, a Nav Bar, and
        a Container.
      </p>

      <hr class="my-4" />

      <h2>Simple Form</h2>
      <form>
        <div class="mb-3">
```

```
<label for="emailInput" class="form-label">Email address</label>
<input
  type="email"
  class="form-control"
  id="emailInput"
  placeholder="name@example.com"
/>
</div>
<div class="mb-3">
  <label for="messageText" class="form-label">Message</label>
  <textarea class="form-control" id="messageText" rows="3"></textarea>
</div>

<button type="submit" class="btn btn-primary">Submit Form</button>
</form>

<hr class="my-4" />

<h2>Standalone Buttons</h2>
<p>Here are other button styles:</p>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

#### Exp 4:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Simple Student Registration Form</title>
  <style>
    body {
      font-family: "Segoe UI", sans-serif;
      background-color: aqua;
      margin: 0;
      padding: 20px;
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
    }
    .container {
      max-width: 600px;
      width: 100%;
      margin: 20px auto;
      background-color: #ffffff;
      padding: 30px;
      border-radius: 10px;
    }
    h2 {
      text-align: center;
      color: #333;
      margin-bottom: 25px;
    }
    label {
      display: block;
      margin: 15px 0 5px;
      color: #555;
      font-weight: 600;
    }
    input[type="text"],
    input[type="email"],
    input[type="password"] {
      width: 100%;
      padding: 12px;
      margin-top: 5px;
      border-radius: 5px;
      border: 1px solid #ccc;
      box-sizing: border-box;
    }

    .error {
      color: #e74c3c;
      font-size: 0.85em;
      height: 1em;
    }
```



```

    margin-top: 4px;
}
.success {
    color: #2ecc71;
    text-align: center;
    margin-top: 20px;
    font-weight: bold;
}
button {
    margin-top: 20px;
    width: 100%;
    padding: 12px;
    background-color: #007bff;
    color: white;
    border: none;
    font-size: 16px;
    font-weight: bold;
    border-radius: 5px;
}
button:hover {
    opacity: 0.9;
}
</style>
</head>
<body>
<div class="container">
    <h2>Simple Registration</h2>
    <form id="registrationForm" novalidate>
        <label for="name">Full Name *</label>
        <input
            type="text"
            id="name"
            name="name"
            required
            placeholder="e.g., John Doe"
        />
        <div class="error" id="nameError"></div>

        <label for="email">Email *</label>
        <input
            type="email"
            id="email"
            name="email"
            required
            placeholder="e.g., john.doe@example.com"
        />
        <div class="error" id="emailError"></div>

        <label for="password">Password *</label>
        <input
            type="password"
            id="password"

```

```

    name="password"
    required
    placeholder="At least 6 characters"
  />
  <div class="error" id="passwordError"></div>

  <button type="submit">Register</button>
  <div class="success" id="successMessage"></div>
</form>
</div>

<script>
const form = document.getElementById("registrationForm");
const successMessage = document.getElementById("successMessage");

form.addEventListener("submit", function (e) {
  e.preventDefault();

  const errors = document.querySelectorAll(".error");
  errors.forEach((el) => (el.textContent = ""));
  successMessage.textContent = "";

  let isValid = true;

  const name = form.name.value.trim();
  const email = form.email.value.trim();
  const password = form.password.value;

  if (name === "" || !/^[A-Za-z\s]+$/ .test(name)) {
    document.getElementById("nameError").textContent =
      "Please enter a valid name (letters and spaces only).";
    isValid = false;
  }

  if (email === "" || !/^\\S+@\\S+\\.\\S+$/ .test(email)) {
    document.getElementById("emailError").textContent =
      "Please enter a valid email address.";
    isValid = false;
  }

  if (password.length < 6) {
    document.getElementById("passwordError").textContent =
      "Password must be at least 6 characters long.";
    isValid = false;
  }

  if (isValid) {
    successMessage.textContent = "Registration successful!";
    form.reset();
  }
});
</script>

```

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

## Exp 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>
      const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

      console.log("Original Array:", numbers);
      const filteredNumbers = numbers.filter((num) => num > 5);

      console.log("After filter() (numbers > 5):", filteredNumbers);
      const mappedNumbers = numbers.map((num) => num * 2);

      console.log("After map() (doubled values):", mappedNumbers);
      const sum = numbers.reduce(
        (accumulator, current) => accumulator + current,
        0
      );

      console.log("After reduce() (total sum):", sum);
    </script>
  </body>
</html>
```

## Exp 6:

```
<!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 getGreeting() {
        return new Promise((resolve) => {
          setTimeout(() => {
            resolve("Hello, world!");
          }, 1000);
        });
      }

      async function runProgram() {
        console.log("Program started... waiting for greeting...");

        const greeting = await getGreeting();

        console.log("Got it!");
        console.log(greeting);
        console.log("Program finished.");
      }

      runProgram();
    </script>
  </body>
</html>
```

## Exp 8:

```
import './App.css';
import { React, Component } from 'react';
class HelloWorld extends Component {
  constructor() {
    super();
    this.state = { name: "World" };
  }
  render() {
    return <h1>Hello, World!{this.state.name}</h1>;
  }
}
const Functional = () => {
  return <h1>Hello from Functional Component!</h1>;
};

function App() {
  return (
    <div>
      <HelloWorld />
      <Functional />
    </div>
  );
}

export default App;
```

### Exp 9:

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

const ChildComponent = (props) => {
  const { currentCount, onIncrement } = props;

  return (
    <div style={{ border: "2px solid blue", padding: "10px", margin: "10px" }}>
      <h2>Child Component</h2>

      <p style={{ fontSize: "24px", fontWeight: "bold" }}>{currentCount}</p>

      <button
        onClick={onIncrement}
        style={{
          cursor: "pointer",
          backgroundColor: "#eef",
          border: "1px solid blue",
          padding: "5px 10px",
          borderRadius: "4px",
        }}
      >
        Increment
      </button>
    </div>
  );
};

const App = () => {
  const [count, setCount] = useState(0);

  const incrementCount = () => {
    setCount((prevCount) => prevCount + 1);
  };

  const resetCount = () => {
    setCount(0);
  };

  return (
    <div style={{ border: "2px solid red", padding: "10px", margin: "10px" }}>
      <h2>Parent Component</h2>

      <p style={{ fontSize: "24px", fontWeight: "bold" }}>{count}</p>

      <button
        onClick={resetCount}
        style={{
          marginRight: "10px",
          cursor: "pointer",
          backgroundColor: "#fee",
          border: "1px solid red",
        }}
      >
        Reset
      </button>
    </div>
  );
};
```

```
      padding: "5px 10px",
      borderRadius: "4px",
    }}
  >
    Reset
  </button>

  <hr style={{ margin: "20px 0" }} />

  <ChildComponent currentCount={count} onIncrement={incrementCount} />
</div>
);
};

export default App;
```



## Exp 10:

Server.js

```
const express = require("express");
const mongoose = require("mongoose");
const path = require("path");

const app = express();

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

app.use(express.json());
app.use(express.urlencoded({ extended: false }));

const MONGO_URI = "mongodb://localhost:27017/myLoginSystemDB";
mongoose
  .connect(MONGO_URI)
  .then(() => {
    console.log("Successfully connected to MongoDB!");
  })
  .catch((err) => {
    console.error("Database connection error:", err);
  });

const UserSchema = new mongoose.Schema({
  email: {
    type: String,
    required: true,
    unique: true,
    lowercase: true,
  },
  password: {
    type: String,
    required: true,
  },
});
const User = mongoose.model("User", UserSchema);

app.get("/", (req, res) => {
  res.redirect("/login");
});

app.get("/login", (req, res) => {
  res.render("login", { error: null });
});

app.get("/register", (req, res) => {
  res.render("register", { error: null });
});

app.get("/dashboard", (req, res) => {
  res.render("dashboard");
});
```

```

});

app.post("/register", async (req, res) => {
  try {
    const { email, password } = req.body;
    const existingUser = await User.findOne({ email });

    if (existingUser) {
      return res.render("register", { error: "User already exists." });
    }

    const newUser = new User({ email, password });
    await newUser.save();

    res.redirect("/login");
  } catch (error) {
    res.render("register", { error: "Server error. Please try again." });
  }
});

```

```

app.post("/login", async (req, res) => {
  try {
    const { email, password } = req.body;
    const user = await User.findOne({ email });

    if (!user) {
      return res.render("login", { error: "Invalid credentials." });
    }

    const isMatch = password === user.password;
    if (!isMatch) {
      return res.render("login", { error: "Invalid credentials." });
    }

    res.redirect("/dashboard");
  } catch (error) {
    res.render("login", { error: "Server error. Please try again." });
  }
});

```

```

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});

```

Views/dashboard.ejs

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```

<title>Dashboard</title>
<script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4"></script>
</head>
<body class="bg-gray-100 flex items-center justify-center min-h-screen">
  <div class="w-full max-w-md">
    <div class="bg-white p-8 rounded-lg shadow-lg text-center">
      <h1 class="text-3xl font-bold text-green-600 mb-6">
        Login Successful!
      </h1>
      <p class="text-gray-700 mb-6">Welcome to your dashboard.</p>
      <a href="/login" class="text-blue-500 hover:underline">Logout</a>
    </div>
  </div>
</body>
</html>

```

Views/login.ejs

```
<!DOCTYPE html>
```

```

<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Login</title>
    <script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4"></script>
    <style>
      .error {
        background-color: #f8d7da;
        color: #721c24;
        padding: 10px;
        border-radius: 5px;
        text-align: center;
        font-weight: 600;
        margin-bottom: 15px;
      }
    </style>
  </head>
  <body class="bg-gray-100 flex items-center justify-center min-h-screen">
    <div class="w-full max-w-md">
      <div class="bg-white p-8 rounded-lg shadow-lg">
        <h2 class="text-2xl font-bold text-center text-gray-800 mb-6">Login</h2>

        <!-- EJS Error Display -->
        <% if (error) { %>
          <div class="error"><%= error %></div>
        <% } %>

        <!-- Form now posts to /login -->
        <form action="/login" method="POST">
          <div class="mb-4">
            <label for="loginEmail" class="block text-gray-700 mb-2">
              >Email</label>

```

```

    >
    <input
      type="email"
      id="loginEmail"
      name="email"
      class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:ring focus:border-blue-
300"
      required
    />
  </div>
  <div class="mb-6">
    <label for="loginPassword" class="block text-gray-700 mb-2">
      >Password</label>
    >
    <input
      type="password"
      id="loginPassword"
      name="password"
      class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:ring focus:border-blue-
300"
      required
    />
  </div>
  <button
    type="submit"
    class="w-full bg-green-500 text-white py-2 rounded-lg hover:bg-green-600 transition
duration-200"
  >
    Login
  </button>
</form>
<p class="text-center text-gray-600 mt-4">
  Don't have an account?
  <a href="/register" class="text-blue-500 hover:underline">
    >Register here</a>
  >
</p>
</div>
</div>
</body>
</html>

```

views/register.ejs

```
<!DOCTYPE html>
```

```

<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Register</title>
    <script src="https://cdn.jsdelivr.net/npm/@tailwindcss/browser@4"></script>
    <style>

```

```

.error {
  background-color: #f8d7da;
  color: #721c24;
  padding: 10px;
  border-radius: 5px;
  text-align: center;
  font-weight: 600;
  margin-bottom: 15px;
}
</style>
</head>
<body class="bg-gray-100 flex items-center justify-center min-h-screen">
  <div class="w-full max-w-md">
    <div class="bg-white p-8 rounded-lg shadow-lg">
      <h2 class="text-2xl font-bold text-center text-gray-800 mb-6">
        Register
      </h2>

      <!-- EJS Error Display -->
      <% if (error) { %>
        <div class="error"><%= error %></div>
        <% } %>

      <!-- Form now posts to /register -->
      <form action="/register" method="POST">
        <div class="mb-4">
          <label for="registerEmail" class="block text-gray-700 mb-2">
            >Email</label>
          >
          <input
            type="email"
            id="registerEmail"
            name="email"
            class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:ring focus:border-blue-
300"
            required
          />
        </div>
        <div class="mb-6">
          <label for="registerPassword" class="block text-gray-700 mb-2">
            >Password</label>
          >
          <input
            type="password"
            id="registerPassword"
            name="password"
            class="w-full px-3 py-2 border rounded-lg focus:outline-none focus:ring focus:border-blue-
300"
            required
          />
        </div>
        <button

```

```
        type="submit"
        class="w-full bg-blue-500 text-white py-2 rounded-lg hover:bg-blue-600 transition duration-
200"
    >
        Register
    </button>
</form>
<p class="text-center text-gray-600 mt-4">
    Already have an account?
    <a href="/login" class="text-blue-500 hover:underline">Login here</a>
</p>
</div>
</div>
</body>
</html>
```